

User's Manual for MX-70 Printer

by
David A. Lien

COMPUSOFT® PUBLISHING
A Division of CompuSoft, Inc.
P.O. Box 19669 • San Diego, California 92119 U.S.A.

This Book was prepared especially for Epson America, Inc. by CompuSoft® Publishing. All rights, domestic and international are reserved by CompuSoft®, Inc. Requests for permission to reproduce or distribute this User's Manual should be addressed to:

COPYRIGHT DEPT
CompuSoft®, Inc.
P.O. Box 19669
San Diego, CA 92119

*Copyright © 1981 by CompuSoft® Publishing,
A Division of CompuSoft®, Inc.
San Diego, CA 92119*

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written permission of the publisher. No patent liability is assumed with respect to the use of the information contained herein. While every precaution has been taken in the preparation of this book, the publisher assumes no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein.

CompuSoft® is a registered trademark of CompuSoft®, Inc.

International Standard Book Number: 0-932760-06-6
Library of Congress Catalog Card Number: M-70726

10 9 8 7 6 5 4 3 2 1

Printed in the United States of America

FCC Compliance Statement

This equipment generates and uses radio frequency energy. If not installed and used properly, that is, in strict accordance with the manufacturer's instructions, it may cause interference to radio and television reception. It has been type tested and found to comply with the limits for a Class B computing device in accordance with the specifications in Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference in a residential installation. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient the receiving antenna
- Relocate the computer with respect to the receiver
- Move the computer into a different outlet so that computer and receiver are on different branch circuits.

If necessary, the user should consult the dealer or an experienced radio/television technician for additional suggestions. The user may find the following booklet prepared by the Federal Communications Commission helpful:

“How to Identify and Resolve Radio-TV Interference Problems.”

This booklet is available from the U.S. Government Printing Office, Washington, DC 20402, Stock No. 004-000-00345-4.

Trademark Acknowledgements

TRS-80 is a Trade Mark of Radio Shack. Centronics is a Trade Mark of Centronics, Inc. Apple is a Trade Mark of Apple Computers, Inc. Microsoft is a Trade Mark of Microsoft, Inc.

A Personal Note From the Author

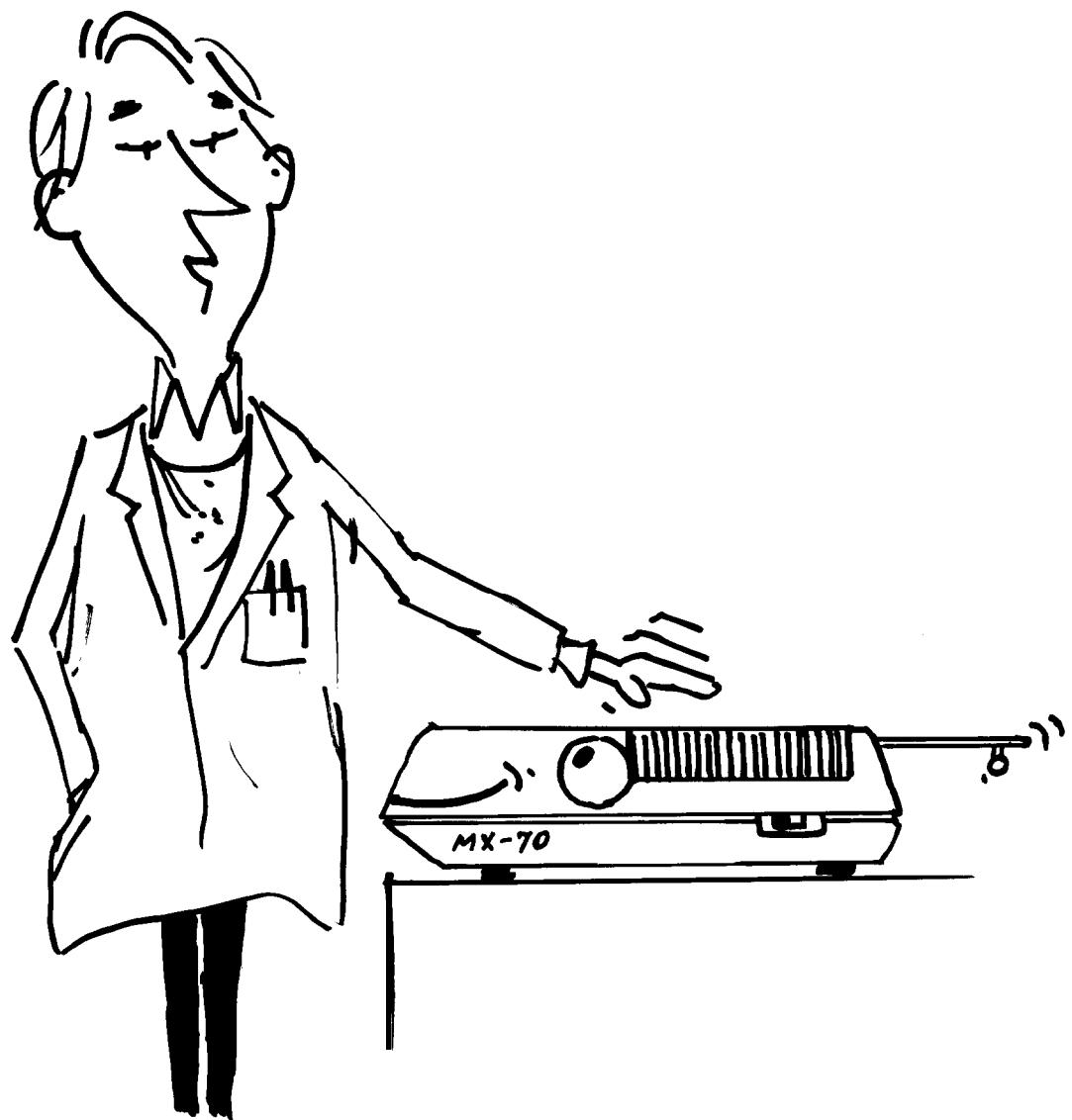
Congratulations on your decision to buy an Epson MX-70 printer! In my opinion, it's the best dollar value in a low cost printer on the market today. Like its big brothers the MX-80 and MX-100, its mechanical features are unsurpassed for the cost. Its electronics features are ideal for the user looking for top value in a printer for general utility use, and high resolution graphics.

***"WHO NEEDS A LEARNERS MANUAL FOR A PRINTER?" The answer -
ALMOST EVERYONE EXCEPT A COMPUTER PROFESSIONAL.***

Today's printers are very sophisticated compared to those of even a year ago. Most are not fully utilized because the instructions are too vague and confusing. We're doing our best to eliminate that problem with the Epson MX-series.

Z encourage you to learn all about your new MX-70. You paid for it. Put it to work.

*Dr. David A. Lien
San Diego - 1981*



MX-70 Table of Contents

FCC Compliance Notice	iii
Trademark Credits	iv
Personal Note from the Author	v
Introduction	ix
Chapter 1: The Starting Line	1
Chapter 2: SenditaMessage	13
Chapter 3: More Print Control Commands	23
Chapter 4: An Introduction to Dot Matrix Printing	31
Chapter 5: Graphtrax II	41
Chapter 6: Advanced Graphics	49
Chapter 7: The Final Push	59
Chapter 8: Using the HIRES Screen Dump Program	67
Appendix A: ASCII Charts for MX-70	79
Appendix B: Control Codes	81
Appendix C: Character Set	83
Appendix D: TRS-80 Differences	85
Appendix E: Use with Atari	89
Appendix F: Use with Apple	91
Appendix G: Special Notes on Other Computers, Languages/Interfaces	95
Appendix H: Technical Specifications	97
Appendix I: Control Circuit Board and Initialization	99
Appendix J: Pinout Chart	101
Appendix K: Parallel Interface Timing	103
Appendix L: Printer Maintenance	105
Appendix M: Schematic Diagram	107
Index	109
Notice	111

USER'S SERVICE MANUAL

WARNING

**High voltage exists inside this unit and
the case should be opened only by a qual-
ified person!**

USER'S SERVICE MANUAL

WARNING

**High voltage exists inside this unit and
the case should be opened only by a qual-
ified person!**

Introduction

“But do I REALLY have to take a course on ‘How to operate a printer’ to use this one?” No - not if you only want to use it for mundane printing. It prints “mundanes” very nicely.

If you want to use the exotic features it offers, approach your new printer in the same way as your first computer - with a healthy curiosity and an open mind. They probably far exceed what you expected from a printer inexpensive enough to call your own.

The MX-70 printer works with virtually any computer properly interfaced to it. This “Learner’s Manual” uses the popular Apple II Plus as its “driver” since the Apple’s high resolution graphics features so closely parallel the printers. What’s taught here applies to all computers, though not all are powerful enough to utilize every MX-70 feature. More on that in good time.

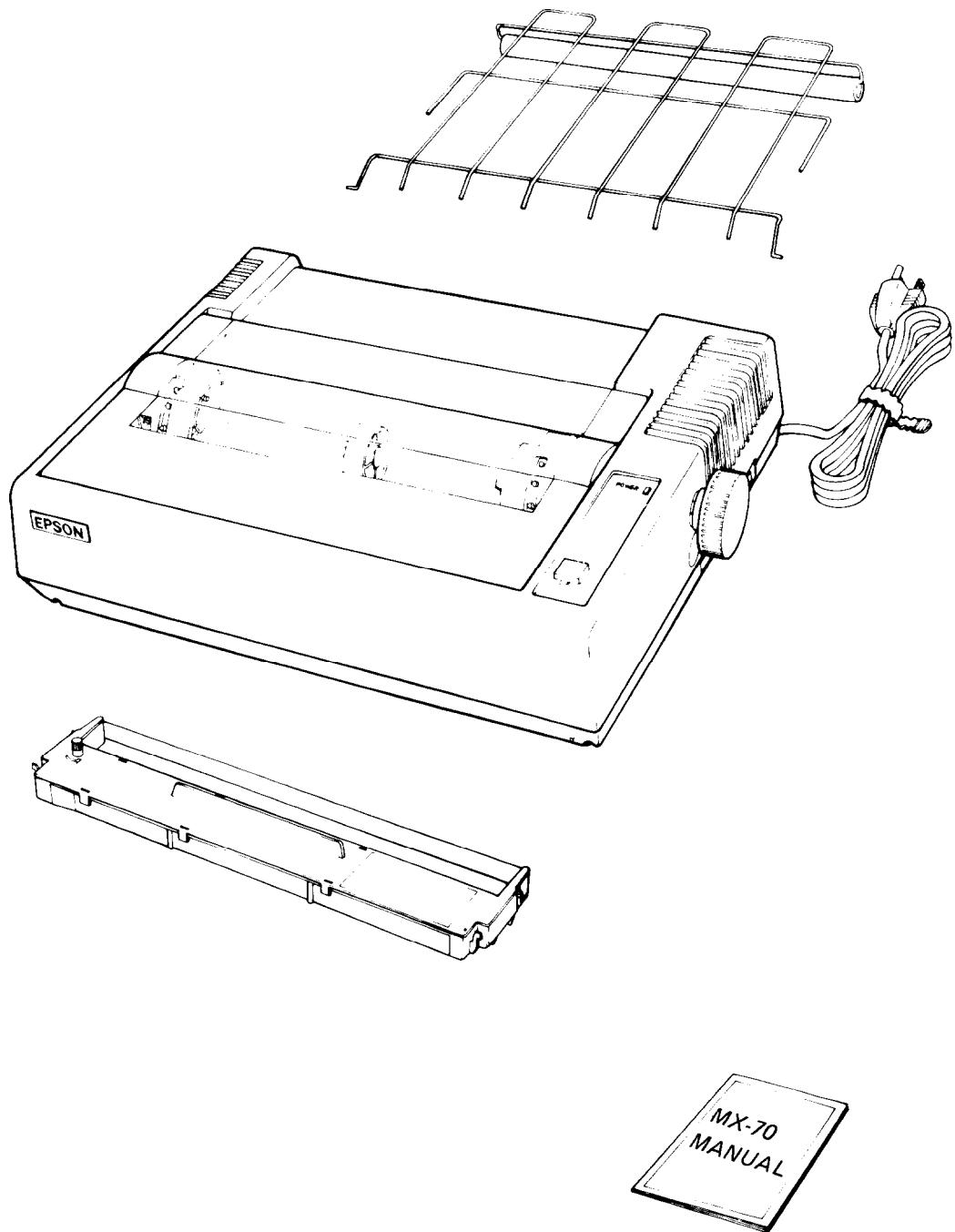
Impatient readers should head for the appendices and usual reference materials at the back. The rest of us will believe that this is no ordinary printer and take the time to learn to use it right - *the first time*.

A good working knowledge of the BASIC* language is all that’s required as we begin, with Chapter 1.

“Users who feel their BASIC skills are a little rusty are referred to the following books by the same author:

The BASIC Handbook (2nd Edition), Encyclopedia of the BASIC Language. 480 Pages. Available through CompuSoft Publishing, Box 19669, San Diego, CA 92119.

Learning TRS-80 BASIC. Available through CompuSoft Publishing, Box 19669, San Diego, CA 92119.



Chapter 1

The Starting Line

This important chapter shows how to unpack, set up, and test your new printer. Read it **before** you get into trouble. The time spent will be your best investment since buying the printer itself.

Counting the Parts

Open the box and carefully and remove the contents. We should find:

1. This User's Manual (obviously . . .)
2. The Epson MX-70 printer
3. A long box containing the ribbon cartridge
4. A wire rack to guide the paper

We also need a cable to connect the printer to our particular computer. Your Epson dealer can provide cables that match the Apple, TRS-80, and other computers. If you use a non-Epson cable, make sure it is wired properly. We have tried many other cables that "should" work, but don't.

The MX-70 uses the Centronics Standard Parallel Interface scheme. Users whose computers require other interfacing schemes should select another printer in the MX series.



Setting it up

Let's first remove the printer lid so we can work without breaking something. Lay the MX-70 flat on a firm surface and raise its lid to the full vertical position (Figure 1-1).

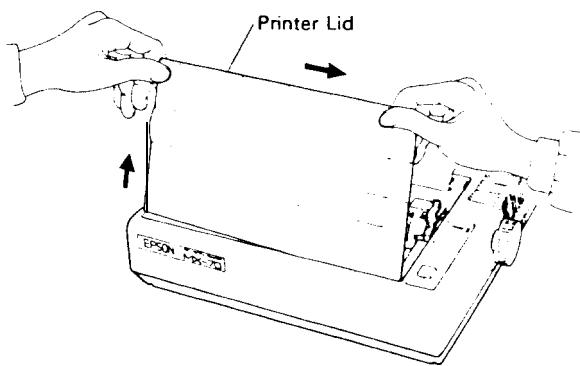


Figure 1-1

With the printer facing you, as shown in Figure 1-2, grasp the lid on its left top side and pull straight up. Very easy. To replace the lid, just reverse the procedure.

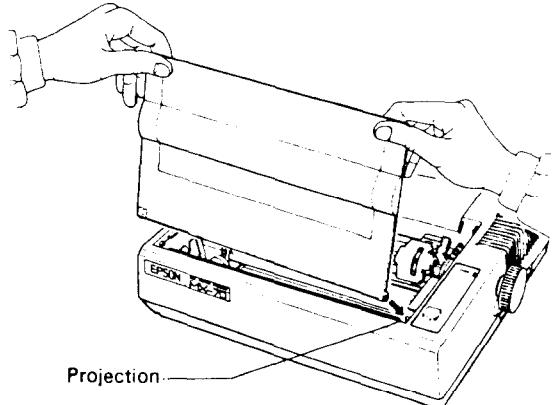


Figure 1-2

Locate the 2 shipping screws, shown in Figure 1-3. These screws are used to protect the MX-70 from damage during shipping.

Shipping Screws

Turn the printer over and lay it on a soft surface. Using a Phillips type screwdriver, remove the shipping screws and save them in case you decide later to lend the printer to Aunt Bernice in Ohio. These screws are in place during shipping to protect the printer from damage and are located as shown in Figure 1-3.

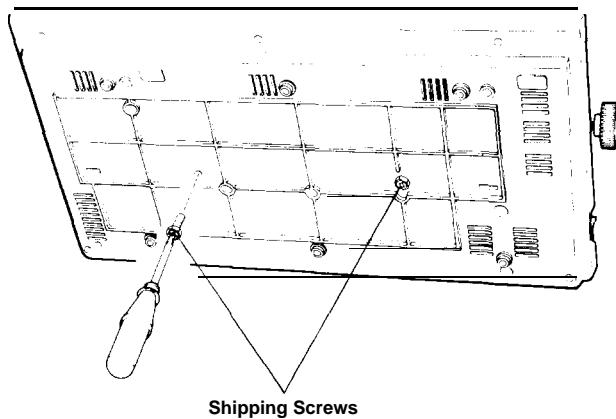


Figure 1-3

Turn the printer back over when you are ready for the next step.

Installing the Ribbon

Position the printer with the Epson label (front) facing you. Locate the paper bail (the moveable metal bar with numbers 1-80 on it - see Figure 1-3) and push it toward the back of the printer, against the metal platen.

Remove the ribbon cartridge from its box and turn the plastic knob counterclockwise so the ribbon is tight. (Figure 1-4A)

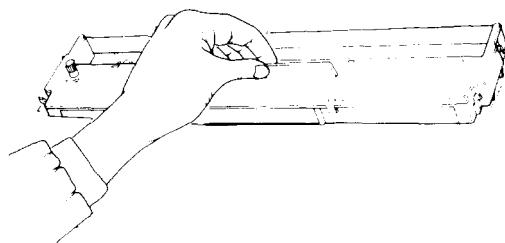


Figure 1-4A

Hold the cartridge by its vertical fin (sounds like a shark). Steer the 4 tabs on its sides into the 4 slots in the printer's metal frame (Figure 1-4B). Press the cartridge firmly into place.

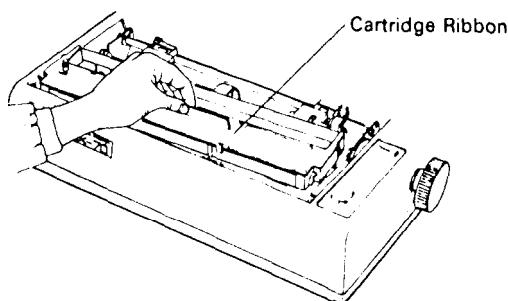


Figure 1-4B

Using a pencil (or your fingers), lift the ribbon onto the slot in front of the print head, as shown in Figure 1-4C. With a little practice, we can do all of the above in one quick operation.

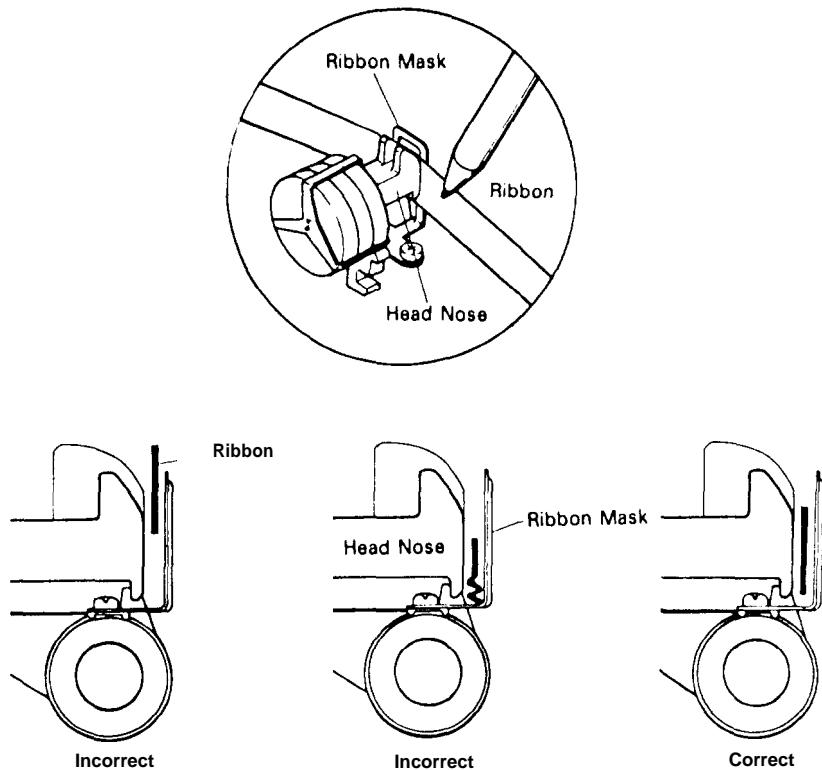


Figure 1-4C

Wind the ribbon tight again with the little plastic knob, and it's all set to go.

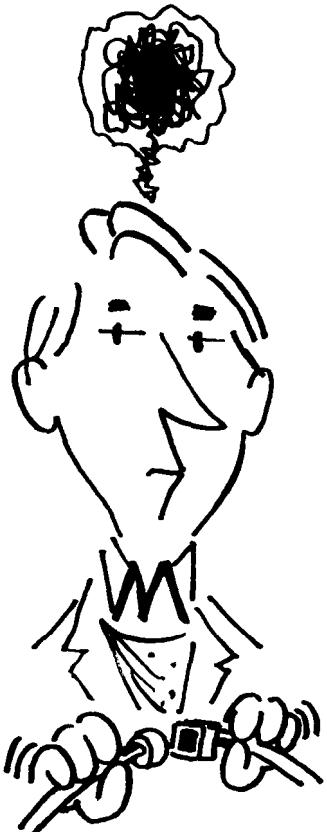
You may notice a sticker with the words EXCHANGE TIMES pasted on the ribbon cartridge. This is a remnant of an early plan to refill the cartridges when the ribbon wears out. That idea was abandoned, at least for the time being.

Matching the Printer to the Computer

At this point, a decision should be made. If the information we need isn't available, we'll decide to defer the decision. Isn't that how all good bureaucracies work?

Some computers, like the Apple, tell the printer to roll the paper up one line each time the print head returns to the left after printing a line. Technically, we say each CR (Carriage Return) is accompanied by an LF (Line Feed).

Other computers, like the TRS-80, rely on the printer to know enough to insert its own LF when it receives a CR. Either way is just fine with the MX-70 since it can be wired either way.



Opening the Case

To locate the wiring terminals, we have to open the case. The lid should still be off. Remove the roller knob by pulling straight out, with firm but steady pressure (Figure 1-5).

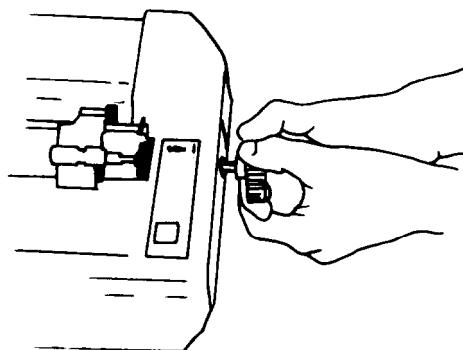


Figure 1-5

Turn the printer upside down on a soft surface. With a Phillips-head screwdriver, completely loosen (but don't remove) all 4 corner screws as shown in Figure 1-6. Place tape over the 4 holes so the screws won't fall out when we tip the printer right-side up again.

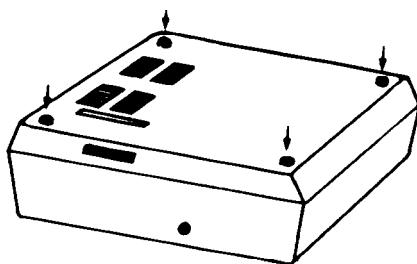


Figure 1-6

Now tip the printer right-side up again.

Gently wiggle the top cover loose. **Wires are booked to it!** We are **not** going to completely remove the cover - only open it enough to gain access to the main wiring board.

Lift the cover up, mostly from the left side, and slide it **gently** to the right to free it from the roller shaft. **Be careful** not to pull the wires on the right-hand side. With just a bit of class we can maneuver the cover so it stands solidly in place, as a sentinel guarding the goodies.

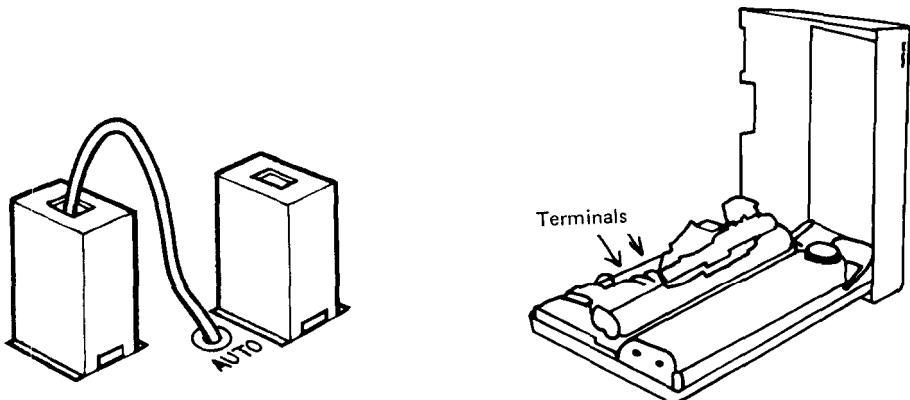


Figure 1-7

Take a minute to gawk at all the stuff in the box. Wow! As Custer said, “Look at all those . . .” well, anyway.

Wonder how they sell it as cheap a they do? (Hope it prints as good as it looks.)

The Epson Connection

Figure 1-7 shows the location of the moveable wire. It is between the 2 “pull top” terminals, and is labeled “auto.” To **release** the wire from one terminal for transfer to the other, pull up **gently** on its colored top. To **lock** a wire in place, insert it fully in a terminal and press the colored top down. Pull gently on the wire to be sure it is securely locked.

If the “auto” wire is connected to the **red** (or orange) terminal, the printer relies on the computer to send its own LF along with CR. That’s what’s needed for Apple-type computers.

If the “auto” wire is connected to the **white** terminal, a LF will automatically be added when a CR is received from the computer. This is what’s needed for the TRS-80 and **most** other computers.

Terrific. So where do we put the wire for **our** computer?

When in Doubt, Bluff

If you have an Apple, move (if necessary) the short white wire located between them to the **red** terminal (as in red apple).

If you have any other computer, hook it (for now) to the **white** terminal.

If we guessed wrong, nothing will be harmed. We'll just have to reopen the case and move it to the opposite one. The results will be known early in the next chapter.

A plastic plate is taped to the access hatch on the inside back of the lid. We can either leave it in place as we replace the lid, tape it to the **outside** of the hole, or add it to our growing collection of printer parts. In theory, the wire could be moved from one terminal to the other through this hatch - but only by a skilled dentist.

If you are **sure** the wire is on the proper terminal, put the printer back together and tighten the screws. If not sure, put it back together and **don't** tighten the screws. The cover goes back on the way it came off - very carefully.



Printer Cable Connection

We're getting closer to the good stuff.

Be sure both the computer and printer are turned off. Connect the printer cable to the printer **only** -not to the computer! It plugs into the only jack on the back of the MX-70 case.

Epson dealers can supply cables to match different computers. Check with your Epson dealer to be sure you have the **right** one. Cables supplied with the computer frequently do **not** work. Do **not** connect the other end of the cable to the computer, yet.

Take your time! Double-check to ensure that the cable is in place **and locked**. It may take a firm push on the connector to secure the locking clips. Without this tight connection, printing may be erratic. Again - leave the **other** end disconnected.

The Paper Rack

The wire rack with the plastic roller tube is a paper separator. It allows the paper to feed smoothly in to and out of the printer, holding it away from the cable.

Chapter 1

To install the rack, simply spring the wires a bit so they pop into the holes in the printer frame, as shown in Figure 1-8.

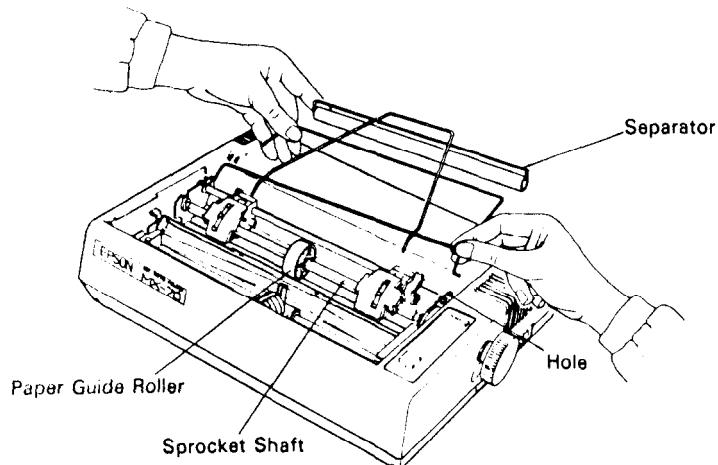


Figure 1-8

Feeding the Paper

The MX-70 accepts pin-feed paper between 4" and 10" wide. Both the left and right hand **tractors** are adjustable to match the spacing between the drive holes. (We never get too far from the land, do we?)

Position the box or stack of paper behind and below the printer. Reliable operation depends somewhat on the weight of the paper keeping itself taut (Figure 1-9).

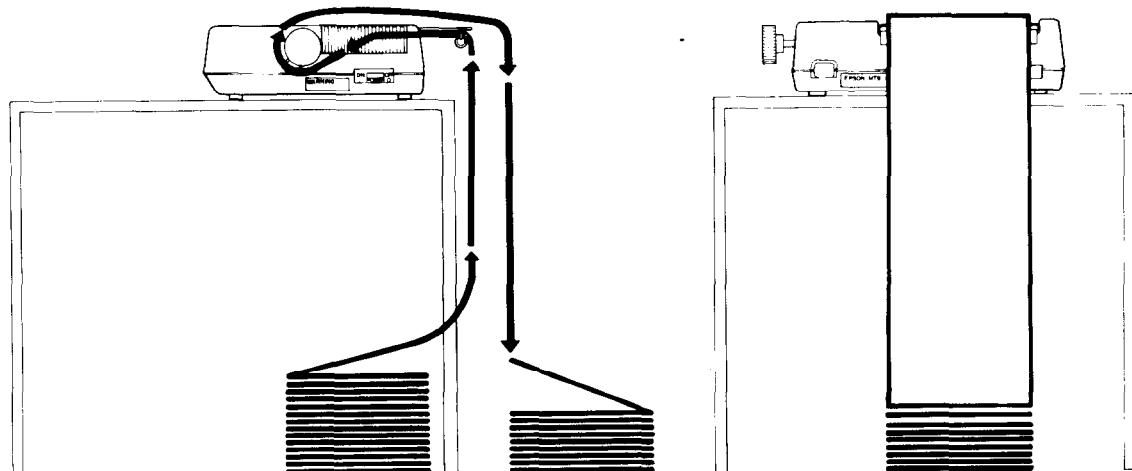


Figure 1-9

Pull the paper bail forward. Open both tractor covers and center the paper guide roller between them (Figure 1-10).

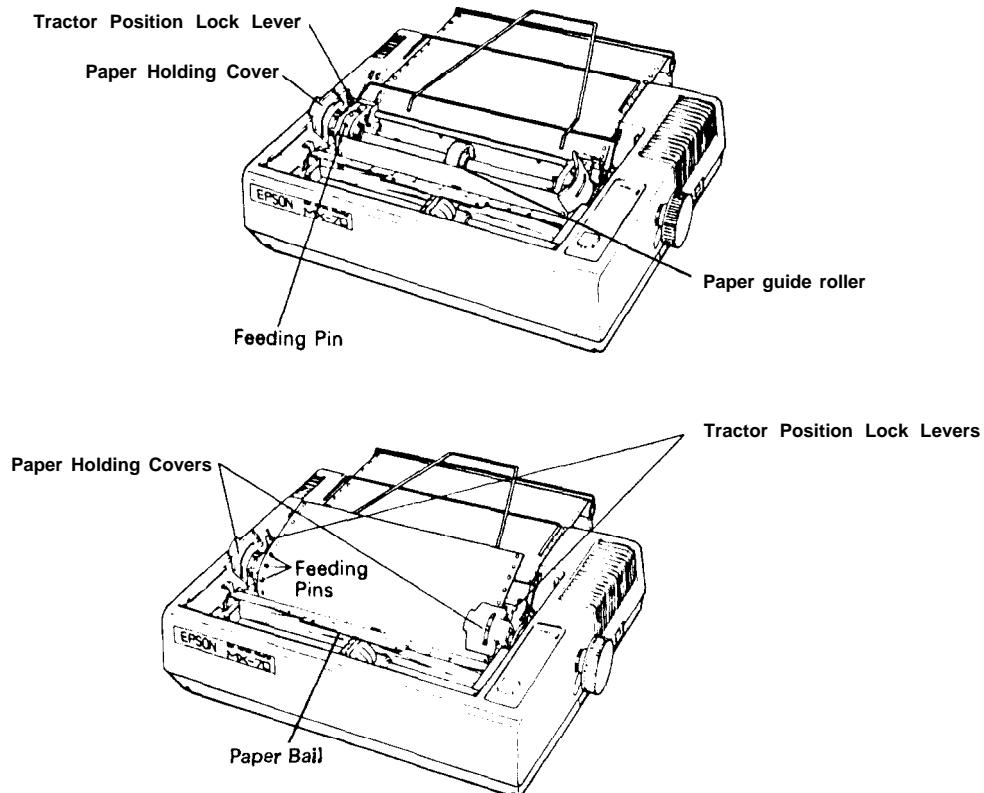


Figure 1-10

Feed the incoming paper **above** the plastic tube but **below** the wire frame, into the slot and right on around to the tractors. Move the tractors as necessary to match the hold spacing in the paper. The tractor position lock levers are shown in Figure 1-10.

Position the paper holes on top of the tractor teeth and close the tractor covers. Adjust one or both tractors so the paper is centered as you wish it, and is held firmly in place. Push the bail back up against the paper.

Roll the paper forward with the roller knob.

The printer moves paper **forward** only, and never looks back. If we **must** turn the roller back manually, the power should be OFF; it helps to pull lightly on the paper.

Pull the bail back out of the way and set the paper so the top of a new sheet is positioned right at the scribe mark on the metal platen (Figure 1-11). Push the bail back up against the paper.

Adjusting for Paper Thickness

The MX-70 can print on all types of pre-printed multiple copy forms and carbons, as well as ordinary printer paper. The 7 position thickness control lever (Figures 1-10 and 1-11) moves the print head closer to or farther from the paper, changing the print quality somewhat. For ordinary, single-thickness paper, start out with the control lever at about the middle position. For multiple copy forms, pull it towards the front of the printer. Push the lever towards the rear of the printer to produce slightly darker print.

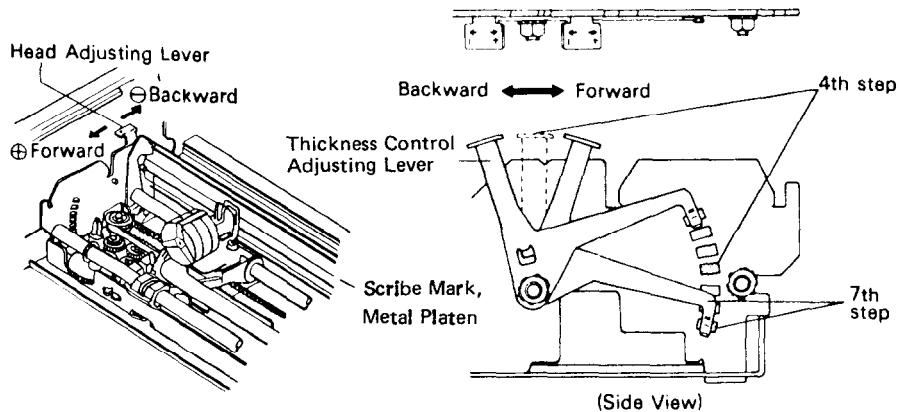


Figure 1-11

Plugging it in

One more item, and we'll be off and running.

Printers sold in the U.S. and Canada are designed for a standard 120V, 60Hz outlet, and have a 3 wire ground plug. **Do not attempt to defeat the grounding.** When you've located a proper outlet, see that the POWER switch on the right-hand side (Figure 1-10) is OFF, then plug the cord in!

At Last

The big moment is here. TURN IT ON!

Mmmm! Look at that! A few startup clicks, a single green light, and it just sits there waiting. The old days were never like this.

Where is all the motor noise? How about some big levers, lights, and alarms? Looks like dynamite in a small silent package!

The POWER light should be lit. Press the FEED button several times and watch it advance the paper. This doesn't look at all complicated.

The Final Checkout

The final part of this checkout takes about 3½ minutes. It's important that we do it; plus it shows us what print characters are available.

Load the printer with plenty of paper. Turn the printer OFF. Hold the FEED button down and turn the printer back ON - **at the same time**.

GO!

WOW! Look at that son-of-a-gun go. Watch the head printing. Move the paper bail out of the way to see it better, but keep your fingers (and hair) out of the hardware.

When the printer stops, advance the paper forward with the FEED button, and tear off the self-test run. Hang it up on the wall as a souvenir.

Whew!

That's enough for this chapter. Take a short walk to vent the exhilaration. All's well and we're on the right track. In the next chapter we'll hook it to the computer, turn it on, and see what happens.



GO !

Chapter 2

Send it a Message

The MX-70 printer is smart. It knows how to follow instructions.

Any Code Devised by Man Can be Broken by Man

Many instructions are sent to the printer. Every letter, number, and other character travels from the computer through the printer cable in the form of a code made up of numbers. We know it as the ASCII code (American Standard Code for Information Interchange), pronounced ASK-key.

Let's take a quick glance at Appendix A to refresh our memory. The decimal number 65 stands for the letter A - etc.

The ASCII code numbers for upper case letters, numbers, and punctuation are pretty well standardized around the world. Unfortunately, the remaining code numbers are used in a variety of different ways, even among manufacturers within a single country.

Besides letters and numbers, we can also send "special" codes to make the MX-70 print wide letters, or enter a special graphics mode that permits us to control where each individual dot is printed. To take advantage of these features, however, the **computer** has to be able to **send** these special code numbers.

As we will see, each with his own computer, not all computers can send all code numbers. With printer technology advancing faster than computer technology, the computer has replaced the printer as the weak link in the system.

The Code Courier

In many cases, the easiest way to send these special codes is to "build" them into the computer program along with its "regular" codes for ordinary letters and numbers. We can do this easily using programs written in the BASIC (or another) computer language. Because of its simplicity and overwhelming popularity, we will do all our demonstrating and learning here in BASIC.

We can also send these special codes at some computers "command level" **before** running a program. A program may even contain codes to change earlier codes, allowing us to print things the way we want them, when we want them.

The route to success in BASIC is via CHR\$ (pronounced Character String). Users with so-called Integer BASIC, which may not support the CHR\$ function, may be restricted to very routine printing. To use the MX-70's exotic



graphic and other features they will have to resort to POKEing codes or some other devious means, and are referred to their own computer's reference manual.

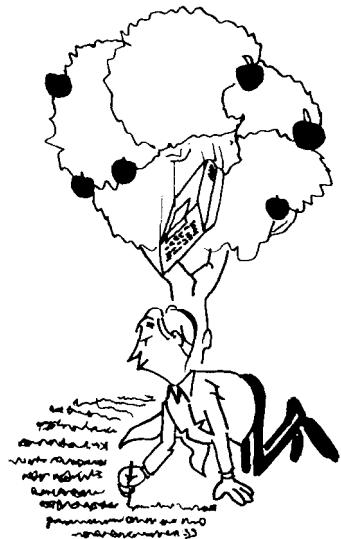
The Philosophy

This is a book about a printer, but since a printer without a computer is worthless, we must also talk about computers. It is still a book about how to use a printer.

We have chosen to write this manual around the Apple II-t computer, using Applesoft BASIC. The Apple features high resolution graphics like the MX-70, so advanced users can create images on their screen and dump them directly to the printer. (The Epson MX-80 printer manual was written around the TRS-80 for similar obvious reasons.) Most other computers can also print HI-RES graphics on the MX-70, even if they can't print those same graphics on their video screen. (Think about **that** one for a minute!)

To meet the needs of the maximum possible number of users, the first 7 chapters assume use of a simple 16K non-disk system. Users with more elaborate systems can either pull the disk controller card and follow along, or see Appendix F for special BASIC disk considerations.

Chapter 8 teaches use of the special disk-based screen dump program (called HIRES), which can be supplied by Epson dealers for Apple computers. It logically assumes that any user who can use that diskette has an Apple disk system with 48K of user memory, and the chapter is written accordingly.



Variations on a Theme by Dartmouth

BASICs do vary. Applesoft BASIC and some others communicate with printers in a distinct way. A command such as PR#1 "hooks" the printer (or whatever

may be connected to Port #1) to the code stream sent to the video screen. PR#0 “unhooks” Port #1 from the video signal chain by sending ALL printing only to the screen.

Most other BASICs use a special word such as LPRINT to send codes directly (and exclusively) to the printer. A separate PRINT statement must be used to send codes exclusively to the video screen.

A few computers even have a **dual** command that allows use of just PRINT, but everything that goes to the screen also goes to the printer port. The computer you use determines which scheme you select.

If you’re not using an Apple, peel and slice the Apple programs in this manual as needed. If you know your own computer, it should be easy. If this causes gastric acidity, I respectfully suggest referring to a copy of the second edition of The BASIC Handbook, by your humble servant.

If you are using an Apple computer, continue straight ahead. If it’s a TRS-80, System 80, PMC-80, or any other machine using a similar Microsoft-type BASIC, GOSUB immediately to Appendix D. Atari users GOSUB to Appendix E.

If it’s any other computer that requires a special interface to drive a parallel printer, GOSUB to any instructions that might come with that special interface kit.

Non-Apple users should ignore the few sections that are clearly written for the Apple, like the following paragraph, unless of course you want to learn something about another computer - which isn’t a bad idea.

The Apple Stem

Apple users verify that the power is OFF, then place the Epson Parallel Interface card in slot 1 (not 0) of the computer’s mother board. Connect the printer cable as shown in Figure 2-1.

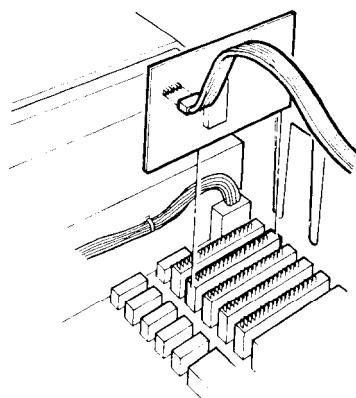


Figure 2-1

There are many parallel interface cards made by other manufacturers. Be sure the card you purchase is designed to work with the MX-70.

Testing the Hookup

The time has come to see if all this hardware works. Whatever your computer is, now's the time to be sure it's hooked up to the printer.

We will soon find out if we hooked the "auto" wire to the correct terminal, mentioned in the last chapter.

Turn on the MX-70 first, then the computer. Type in the following software:

PR#1	(for Apple users only, to "switch in" the printer. We will leave it tied to the screen for this entire chapter.)
10 FOR P = 1 TO 24	(TRS-80 and most other Microsoft
20 PRINT P	BASICs use LPRINT in Line 20)
30 NEXT P	

and RUN.

If the printing is single-spaced (6 lines per inch), skip to the next paragraph. If the printing is double-spaced, go back inside the printer to switch the "auto" wire to the other terminal. (Don't forget to turn **off all** the power first!) Remember:

RED CAP - NO LF ADDED
WHITE CAP - ADDS LF TO EACH CR RECEIVED

If nothing printed, the problem is probably in the cable. Many connectors are wired differently than the Epson cable. See Appendix J for the correct MX-70 connector wiring.

When All Systems Are "Go"

When all is well, type in this short demonstration program, but DO NOT RUN it, yet:

- TRS Model I users - see Appendix D for help with line 30.
- Atari users - see Appendix E for help with trailing semicolons.

```

10 PRINTTAB (14) ; "GREETINGS FROM THE GRAPHIC"
20 PRINTTAB (22); CHR$ (14); "MX-70"
30 PRINTTAB (22) CHR$ (27) "K" CHR$ (60) CHR$ (0);
40 FOR I=1 TO 25 STEP .4
50 PRINT CHR$ (2^INT(3.4*(SIN(I)+1)));
60 NEXT I : PRINT
70 PRINTTAB (20) CHR$ (98) CHR$ (121) CHR$ (14)' EPSON'

```

Time Out For Emergency Training

CAUTION: If you make a typing error that causes the program to crash, or look weird, be sure to turn the printer OFF then on again before running the corrected program. The faulty program may have sent an unwanted code "down the line." It may even have sent something unpatriotic like "don't listen to the computer." The printer will reset itself to "normal" by simply turning it OFF and ON. (Yep -just like they had to do on the first space shuttle flight).

It may even be necessary to shut the computer down cold, and start over from the beginning. The printer has its own internal "computer," and the two computers talk to each other. If one decides to throw a temporary snit, we have to get in between them and cool things down. It doesn't happen often, but as with any computer, a glitch on the power line or a static electricity discharge can cause all sorts of heartburn. Best to know how to handle it when it comes.

Operator Now Certified

Now RUN.

There it is! We finally strapped these two pieces of high technology into harness and got us a real convoy.

GREETINGS FROM THE GRAPHIC
 M X - 7 0
 b y E P S O N

Figure 2-2

RUN it again. This time pull the paper bail forward to get a better look at the print head in action.

There is a lot happening, so we'd better dissect the program (like a frog) to learn all about it. Type

LIST (TRS-80 etc. use **LLIST**)

to LIST the program on paper.

Push the FEED button to roll the printout past the lid. Tear it off and keep it handy so we can study the program as we go along.

SAVE the program on tape (or disk) to avoid having to type it in later.

Line by Line

We're going to type the last program back in, a line at a time, analyzing it as we go.

An alternate strategy is to load the entire program back in from disk or tape, then delete all the lines except the ones being studied. A second alternative is to leave the program in, inserting a REM before each line, removing the REMs as we progress. A third alternative is to leave the program intact and temporarily insert an END at Line 15, shifting it down the program as we progress.

Take your pick!

Line 10 is straightforward. It PRINTed, the "greetings," starting 14 spaces from the left. Type it in and RUN, watching the head action as it prints.

GREETINGS FROM THE GRAPHIC

Figure 2-3

Well, that was rather "pedestrian."

Line 20 has 3 PRINT statements on the same line. They are separated by semicolons, which are usually optional. The first PRINT statement TABs the print head over 22 spaces, the second sends one of those special CHR\$ (character string) codes we mentioned earlier, and the third prints "MX-70."

CHR\$ (14) stands for: print DOUBLE WIDTH.

Appendix B contains all the special codes, often called CONTROL CODES.

Not all codes actually print on either the screen or the printer. Most special or “control” codes don’t really PRINT anything, even though we must precede them with a PRINT to “push them down the line” to the printer.

For example, type the following at the command level:

```
PRINT CHR$(14); "DOUBLE WIDE"
```

and see the words appear in double width (not double spaced).

```
D O U B L E   W I D E
```

Figure 2-4

LIST the program to paper again. Notice that the double width feature is no longer switched on. The message: EACH TIME WE WISH TO PRINT SOMETHING IN THE DOUBLE WIDE MODE, WE MUST PRECEDE IT WITH CONTROL CODE 14.

When that LINE is finished printing, the double width feature is **automatically turned off**.

Now that we understand line 20, let’s add it in and RUN our 2-line program.

```
GREETINGS FROM THE GRAPHIC
M X - 7 0
```

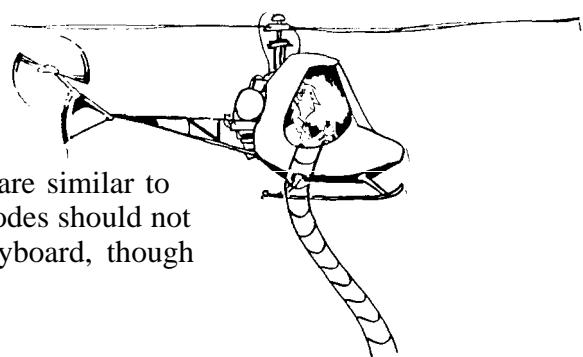
Figure 2-5

Pretty nice - eh? If you wish, do another LIST. Look at the printout to feel confident that DOUBLE WIDTH really got switched off.

The programming in line 30 is a bit more exotic. It consists of only 1 PRINT statement to “push” 1 TAB, a control character, a “K,” and two ASCII characters to the printer. The control code 27 stands for “ESCAPE,” which forces us to GOSUB to another topic.

The Great Escape

The MX-70 recognizes a few so called “Escape Codes.” They are similar to (and really part of) the control codes under discussion. Escape codes should not be confused with the escape KEY which may be on your keyboard, though they are shirttail relatives.



Chapter 2

We send the special code CHR\$ (27), which means “<Escape>,” down the line immediately preceding a control code that needs that <Escape>. These special “code pairs” are logically called “Escape codes.” We can either build them into a BASIC program or send them from the command level, as we did earlier with the simple stand-alone control codes. We will learn to use each “Escape code” as it is needed.

Our First Escape Code

<Escape> “K” stands for KICK it into the GRAPHICS MODE. As part of the graphics mode series of commands we must tell the printer how many columns of graphics we intend to print. The next two codes, CHR\$ (60) and CHR\$ (0) tell that tale:

CHR\$ (60) means “expect 60 characters.” Any additional ones will be printed as text. If less than 60 are sent, the printer will just sit and wait until that quota is satisfied.

CHR\$ (0) tells the printer that we decline to send another 256 characters. If that latter point seems a bit obscure, don’t worry. We’ll get a more detailed explanation of what that means and plenty of practice in the graphics chapter. Just follow along with the act, for now.

Model I TRS-80 users note: the Model I cannot send a CHR\$ (0) reliably to the printer. Try CHR\$ (2) instead or see Appendix D for special instructions to deal with this problem.

Return

RETURNing from our GOSUB, type in line 30, but don’t RUN it, yet. We have to create some graphics to send at the same time.

```
30 PRINTTAB (22) CHR$ (27) "K" CHR$ (60) CHR$ (0);
```

Lines 40 and 60 are the loop that sends the 60 characters from line 50 to the printer. Actually, the loop sends a few more than 60, (2510.4 = 62.5), but we told the printer to only accept 60, so it ignored the “overage.”

Add these lines in preparation for the big one.

```
40 FOR I=1 TO 25 STEP .4
60 NEXT I : PRINT
```

Line 50 actually sends the messages down the line to the printer. The expression inside the CHR\$ function looks complicated, but that's just to keep your attention. Observe its form, but ignore its contents. We'll have to study the whole manual to find out how to generate such a pretty line of graphics.

Add line 50:

```
50 PRINT CHR$ (2^INT (3.4*SIN(I)+1)));
```

and RUN.

Isn't that slick? It gets even better when we understand how to design our own.

The Finale

Only 1 more easy line. It's downhill now.

Line 70 starts out with a TAB and two character codes. It's back to the ASCII chart, Appendix A to see that 98 and 121 are the codes for lower case b and y. We've already used CHR\$ (14) and seen that it puts the printer in double width mode.

Add line 70:

```
70 PRINTTAB (20) CHR$ (98) CHR$ (121) CHR$ (14)" EPSON"
```

and RUN. Very nice.

This has been a feature-packed chapter, but it gave us a good introduction to some varied and powerful capabilities we can put to work. Better take a break before tackling Chapter 3. It wouldn't hurt to go through this chapter once more before moving on, now that we know the plot.

Print Command Codes Learned so Far

CHR\$ (14) - Turns ON double-width characters.
(Goes OFF by self after end of line)

Special ‘Escape’ Codes

CHR\$ (27) - The BASIC ASCII combination for “Escape.”

(Special code used with letter codes)

“K” - The letter “K.” (Kicks ON graphics mode
when preceded by Escape.)

Example: PRINT CHR\$ (27) “K”

Chapter 3

More Print Control Commands

That last chapter was a heavy one, but it gave us a good overview of many MX-70 features. In this chapter we'll explore a few of them in more detail.

As before, CHR\$ is the magic wand. Continue to refer to Appendix B as needed to help maintain perspective on what we're doing and where we're going.

Double the Pleasure

There is an easy way to print 2 sizes of characters on the same line. We simply switch back and forth between single and double WIDTH. The result is quite impressive.

Type in this simple program:

```
9 PR #1          (Apple only. Apple disk uses:)  
                 (9 PRINT CHR$ (4) "PR# 1")  
  
10 PRINT TAB(10) ;      (TRS type uses LPRINT)  
20 PRINT CHRS(14) "DOUBLE "CHRS(20)" WIDTH CHARACTERS";  
30 PRINT "ADD" CHRS(14)" EMPHASIS!"  
39 PR #0          (Apple only. Apple disk uses:)  
                 (39 PRINT CHR$ (4) "PRO")
```

and RUN.

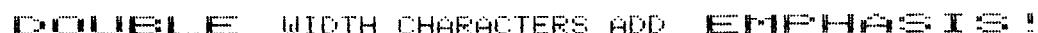


Figure 3-1

Note carefully where the spaces are positioned in lines 20 and 30. Now study the spaces in the printout.

When a space is placed after an ASCII 14, (double width, or CANCEL single width), that space is printed as 2 spaces. If placed after ASCII 20 (single width), it takes up only a single space.

Don't forget, double WIDTH mode is automatically turned off at the end of each line, unless that line ends with a semicolon (;). If we want to print a very long line consisting entirely of double-width characters, the program lines holding the parts of that message must each have a trailing semicolon.

Double-width is quite unique but loses its visual impact if used all the time. We get "bumped" out of it automatically at the end of each line and have to tell the printer each time we actually need it.

Establishing the Facts

The standard "letter size" piece of paper in most parts of the world is 11" long. A maximum of 10" is usually used for printing, the remaining inch divided between the top and bottom margins. "Legal size" paper is 14" long, of which no more than 13" is used. Both sizes are 81/2" wide, with no more than 8" used for printing.

Standard typewriter or printer spacing is 6 lines per inch, or 66 lines per 11 inch page (78 lines for a legal length page). A new line is started each 1/6".

As we've already seen, standard character spacing is 80 characters per line, or 10 per horizontal inch.

Having reviewed and documented these dull facts and traditions, we're pleased to learn that with simple software commands we can change many of these "standards," customizing them for our needs.

So Much for Tradition

Load the printer with plenty of paper as we are going to use a pile of it. Chalk the cost up as tuition.

First, let's see what is really meant by page length. Turn OFF the printer and adjust the paper so the perforation between two sheets is near the scribe mark on the metal impact platen. Let's remove the lid for the rest of this chapter since we'll be peeking at the platen often (Figure 3-2).

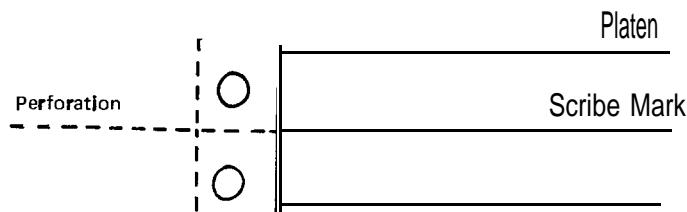


Figure 3-2

It may take a little practice to get the paper positioning perfect. We need to be able to do it right, especially for matching lines and boxes on pre-printed forms such as invoices, bills of sale, purchase orders and the like.

Some users find it easier to position the paper in the ball park, then finish positioning by reference to the match mark on the pin feed holder, one of the tractor pins, something else that moves with the paper, or some point over which the paper travels. Practice will make perfect.

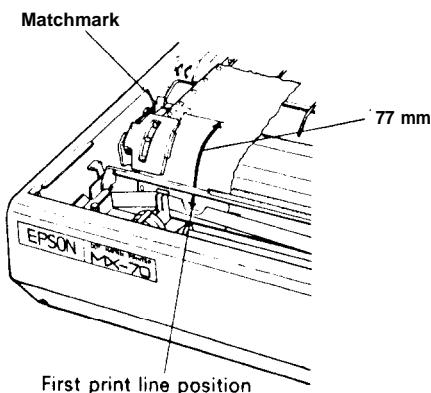


Figure 3-3

Once the paper is in position, cycle the printer OFF, then on again. This electronically sets the top of form memory (known in the biz as TOF) at the current paper position. Despite what follows, the printer always remembers where the top of the next sheet of paper is, even if WE don't. It has a memory like an elephant.

Type in this new program:

```

9 PR #1                               (Apple only)
20 FOR N = 1 TO 66
30 PRINT N
40 NEXT N
49 PR #0                               (Apple only)

```

and RUN.

Well, that was singularly unexciting. What did it tell us? Where is the top of the next form?

1. It told us that the paper length is EXACTLY 66 rows. No more and no less.
2. Since the numbers all line up in a nice column, and there's no vertical spacing between them, the printer does not automatically insert margins at the top or bottom of a page.

The Form Feed - in Basic Software

Let's modify our program to print only 10 lines, then have it automatically roll the paper up to the next TOE CHR\$ (12) means "Feed Form to TOF."

Make the program read:

```
9 PR #1                      (Apple)
20 FOR N = 1 TO 10
30 PRINT N
40 NEXT N
45 PRINT CHR$(12)            (TRS use 12+128=140)
49 PR #0                      (Apple)
```

and RUN.

CURSES! (the villain said). We overshot the top of the next form by one lousy line. Did you look at it?

What went wrong? All we added was a simple Form Feed in line 45. Where did that extra line feed come from? Hmmmm? (Knit 1, Pearl 2.)

That's right! Every PRINT statement automatically sends a Line Feed to the printer. We forgot to include a trailing semicolon in Line 45 to suppress it.

Sounds simple enough. Let's change Line 45 to:

```
45 PRINT CHR$(12);
```

and RUN again.

Time Out for Controlled Confusion

We are genuinely on the horns of a dilemma.

Since the Apple PR#0 disconnects the printer **before** the final Line Feed form line 45 is executed, Apple users are seeing different results than everyone else and are wondering what all the shouting is about. They don't see any problem, but for the wrong reason!

Apple users replace PR#0 in line 49 with END, then RUN and see what happens.

Get the idea, now? It overshot by one line again.

Let's all read the following "Philosophy" carefully. It is absolutely vital that we understand the concept of the "endless loop" to use any printer for serious business type applications.

A Little Homespun Philosophy

Having discovered the problem, and it is a very real and vital one, let's analyze the needs of the average user of the Form Feed (FF) command. Typically, he is a businessman sending out a long string of bills. Or, perhaps it's an organization mailing a stack of letters, or printing address labels. In any case, what happens is repetitive, with perhaps certain places on pre-printed forms filled in by the printer.

This means the printer runs continuously. The program will certainly not stop after each form is printed. The program printing all these forms will be locked in a continuous loop, READING in and processing new information from disk, tape, data lines, or even from the keyboard, then printing it in a specific format on the forms.

Understanding and appreciating that concept is vital to understanding how FF is used.

Now, if we can **delay** execution of the "extra" Line Feeds caused by sending form control commands, we can effectively forget them. Sort of like taxes. Delay them long enough and they don't matter.

We learned long ago in our study of Elementary BASIC that a semicolon (;) at the end of a PRINT line temporarily suppresses its Line Feed and Carriage Return. But when the program comes to its END, the suppressed LF/CR catches up. What do you suppose would happen if we put our PRINT CHR\$ (12) Form Feed command into the continuous loop along with the regular print statements? You got it!

NOTE: The suppressed LF/CR's don't just keep adding up and dump out at the END, spilling all over the floor. Only 1 of them can hang over our head, awaiting disposition.



Charge!

Now we can modify our resident program so it can't END without human intervention:

49 GOT0 20

and RUN. Forget about the paper! If you have to fuss, worry about the medfly.

Now we are all on the same channel, and the printer hasn't forgotten where the Top Of Form belongs.

When the floor is strewn with paper and the point has been made, you may hit CTRL-C, (or BREAK) and let a silent smile creep across your ugly puss. Success is sweet, eh Crock! (CRUNCH!)

Onward

Suppose we're not using full 66 line forms. When's the last time you got a statement on a sheet of paper that big? Oh really? And it started out "Greetings from the President . . ."? Well, anyway.

Let's suppose we're printing on forms that are only half that size, 33 lines per sheet. How do we set the Form Feed to automatically go to a form that isn't 66 lines long? Tho't you'd never ask. (Tho't we'd never get here!)

<ESC> "C" sets the scene for us to set the form length. Add line 10:

```
10 PRINT CHR$ (27) "C" CHR$ (33);
```

<ESC> "C" followed by CHR\$ (##) establishes how many lines are to be on the form - the Form Length. It also resets the Top Of Form to the current print head position, so be careful when using it. The form length can be from 1 to 127 lines long. We will set it to 33. The semicolon at the end of Line 10 delays the inevitable LF until the program ends.

Set the paper to the top of a sheet again and RUN for several sheets. Remarkable! All this fussing and stewing really does pay off.

There's not a lot more to say about FF and LF. It's easy to let them draw us into trouble, yet the trouble is so easy to avoid. If the very concept is giving you problems, go back and restudy the HOMESPUN PHILOSOPHY, then start this chapter over again from the beginning.

Horizontal Tabs

Type in this simple NEW program:

```
9 PR #1                      (Apple)
10 FOR N = 5 TO 75 STEP 5
20 PRINT TAB(N); N           (TRS uses LPRINT)
```

```
30 NEXT N
39 PR #0
```

(Apple)

and RUN.

OK gang. It's different strokes for different folks. Users who can see their printer merrily printing numbers out to column 75 can give their computer a little pat on the - ah, keyboard.

Some TRS-80 users may see the printer tab over to 60, then the carriage returns to print position 0 and continues printing numbers. This is a computer software deficiency - it's not sending the right messages to the printer. Very common. TRS-80's released before Spring 1980 do not send TABs greater than 64.

Apple II on the other hand does not tab correctly past column 40. To prove the point, Apple users change line 20 to:

```
20 PRINT TAB(N); N;
```

and RUN.

It turns out that TABs in BASIC are designed for the 40 column Apple screen. TABs past the 40 column limit are treated as strings of blank spaces. There are 50 blank spaces between the 45 and 50. What's this business about an Apple a day?

There are several ways around the problem. One is to TAB over with PRINT statements containing blanks between the quotes. A nicer way, (not available on the Apple II) is to use STRING\$. See The BASIC Handbook for details.

The point of this program was to show that the MX-70 will accept horizontal TABs from BASIC, and help us determine our own computer's capability to send those TABs.

Code Summary

```
CHR$ (14) - ENTER DOUBLE WIDTH MODE
CHR$ (20) - EXIT DOUBLE WIDTH MODE
CHR$ (12) -FORM FEED (USE 140 FOR TRS-80)
PR# 1 - ENABLES PRINTER FROM APPLE BASIC PROGRAM
PR# 0 - DISABLES PRINTER FROM APPLE BASIC PROGRAM
PRINT CHR$ (4) "PR#1" - APPLE DISK ENABLES PRINTER
PRINT CHR$ (4) "PR#0" - APPLE DISK DISABLES PRINTER
CHR$ (27) "C" CHR$ (##) - ESTABLISHES FORM LENGTH
```

1 <= ## <= 127

Chapter 4

An Introduction to Dot Matrix Printing

(Subtitled, “Does a Dull Title Make a Topic Dull?”)

My Computer Can’t Do High Resolution Graphics!

Correction, your computer **can** do high resolution graphics - on the MX-70 printer. Any computer that has the BASIC CHR\$ function and the correct computer/printer interface can punch out super graphics on the MX-70. It's as easy as sending CHR\$ (0), . . . through CHR\$ (255).

Unfortunately, that may not be as simple as it sounds. If you are fluent in Binary math, you know it takes 8 bits to count from 0 to 255 to specify the above 256 codes. (2 to the 8th power is 256.) (If you aren't fluent in Binary math, it still does. Hang in there.)

Many computers and printer interfaces only support 7 bits of code data, hence, transmit to the printer only the ASCII code numbers between 0-127. At this writing, Epson's Apple parallel interface card is in this category since Apple BASIC does not control the eighth bit. It is quite possible to drive the MX-70 nicely with only code numbers from 0 to 127, but certain things are more difficult.

How Dot Matrix Printing Works

To really understand **dot matrix printing**, we have to understand how the print head works. It contains 9 “wires” or “pins” positioned one above the other, as seen in Figure 4-1.

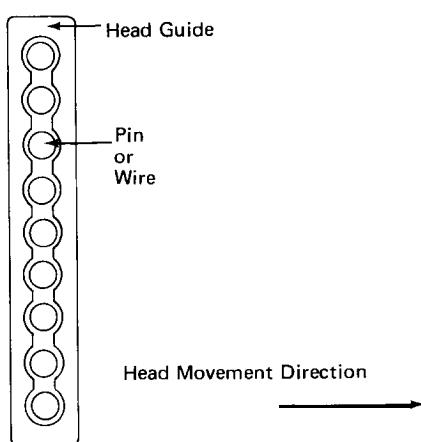
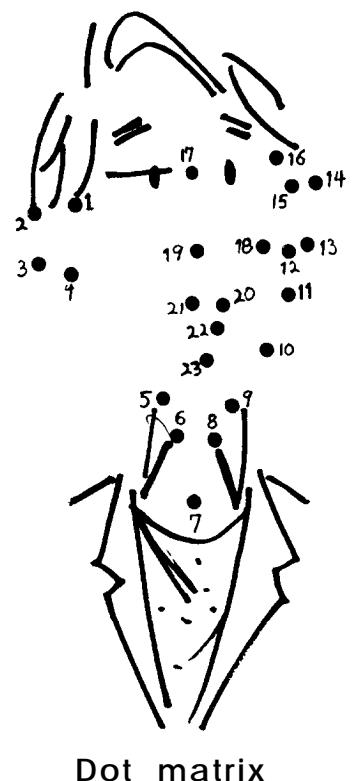


Figure 4-1



Each pin is driven by a “gun,” or electromagnet, which “fires” when told to do so by the electronics inside the printer. The printer electronics responds to those ASCII code instructions we send from the computer in the form of “ASCII numbers.”

For example, to print the letter “I.” wires 1 and 7 are fired first. They hit the ribbon, which makes marks on the paper where the left top and bottom of the letter “I” should be.

The head then shifts over a bit and wires 1,2,3,4,5,6 and 7 all fire at once, printing the center of the letter “I.” One more shift and one more shot of needles 1 and 7 and the letter ‘I’ is complete. (Figure 4-2).

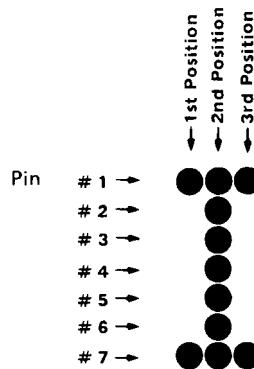


Figure 4-2

This is called “dot matrix” printing. Dots are printed according to a pre-designed “matrix” or “grid” system, where each letter, number, and punctuation mark is formed by an arrangement of dots. As we have seen, this complex printing process takes place **very** quickly.

The pins fire only in groups or clusters. The firing patterns are already programmed inside the printer, matching the alphanumerics (letters and numbers) and symbols found in our ASCII charts. No, we cannot fire an individual needle yet, but that’s what all this is leading up to.

Appendix C shows all our alphanumeric characters constructed within a 7 dot high by 5 dot wide matrix. Every letter, number, and punctuation character is designed to fit within that box.

Notice that the sixth column is always empty. In fact, it isn’t even shown. It’s reserved for horizontal spacing between characters.

Our smallest alphanumeric character, the period, requires only 4 dots in a 2 by 2 grid. Our highest “resolution” in the non-graphic mode is therefore 4 dots.

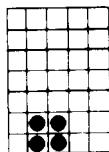


Figure 4-3

For a complete listing of all the standard MX-70 characters, look at the self-test printout from Chapter 1. It's right there, hanging on your wall, isn't it?

How it all Lays out on Paper

We have seen that all the characters are a uniform 7 dots high. What Appendix C doesn't show is that the spacing **between** the lines is a fixed 5 dots. This makes each text line a total of 12 dots high.

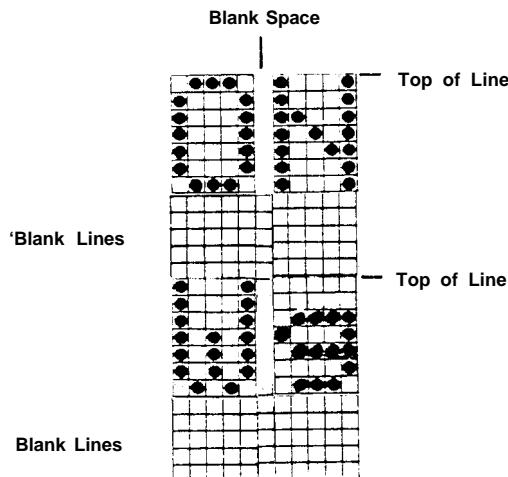


Figure 4-4

Here's another way of looking at it. There are 12 dots from the top of one row of text to the top of the next row. Since the spacing between each pin on the print head is $1/72"$, each line requires $7/72 + 5/72 = 12/72$, or $1/6$ of an inch. 6 lines per inch. No - it isn't just a coincidence.

So what does all this higher mathematics have to do with graphics? Just this. If we are to create high-resolution graphics on the MX-70, we have to eliminate the “dead space” between letters by changing the automatic line spacing to some value less than 1/6 inch. We certainly don’t want a blank space 5 dots high between our rows of graphics!

The Line Spacing is Variable

Now that we understand the dot matrix concept and standard 12 dot “top-to-top” vertical spacing, let’s learn how to change it to suit our specific needs.

Type in this NEW program:

```
9 PR #1          (Apple)
20 PRINT "LINE ONE"
30 PRINT "LINE TWO"
40 PRINT "LINE THREE"
49 PR #0          (Apple)
```

and RUN.

```
LINE ONE
LINE TWO
LINE THREE
```

Figure 4-5

Nothing new here. Just like we’re used to.

Let’s all add line 10:

```
10 PRINT CHR$ (27) "A" CHR$ (12)      (or 12 + 128 = 140)
```

and RUN.

Same as before. Nothing changed. Then what is line 10 all about?

Line 10 contains the following 2 distinct commands:

1. An <ESC> "A." It opens Pandora's box to let us set the vertical line spacing accurate to a single DOT - 1/72nd of an inch. Mercy!
2. A CHR\$ (##). We can specify how many dots we wish to roll down between the top of one line's printing and the top of the next. The number can range from 0 to 85. We picked 12, since that is the default mode. TRS-80 fans and some others may have to use CHR\$ (#+ 128) just to be on the safe side.

Whatever new line spacing we specify is in effect as soon as line 10 is received by the printer.

To dramatically illustrate the possibilities this feature opens up: DELETE lines 30 and 40, and change the following lines:

```
5  S = 1          ' 1  DOT      (TRS use S= 129)
9 PR  #1          (Apple)
10 PRINT CHR$ (27) "A" CHR$ (S)
20 PRINT "LINE ONE",S      (TRS use S-128)
49  S = S + 1 : GOTO  10
```

Line 5 starts us off with a dot spacing of 1/72".

Line 10 brings in the dot number "S" as a variable.

Line 20 prints the dot spacing in 72nds of an inch.

Line 49 adds 1 to the dot spacing, and we do it all again.

RUN until the dot spacing reaches at least 24, then <CTRL-C> or <BREAK> to stop execution.

WOW!

But What do it Mean, Boss?

Well, Rochester, it's like this.

Where the number "1" is printed, the tops of the lines are only one DOT apart (1/72"), instead of the usual 12 dots. That's why the letters are printed all over each other.



Check the printout and find where spacing is 12. Doesn't that look more like what we're used to?

Now look at 24 dot spacing. Aha! It's double spacing. By sending the high-powered command in line 10, we can plug in our own value of S and make the printer give us just about any spacing we want between lines (up to 85). And, we can do it either inside a program or at the command level. That ought to kick the old mind into overdrive!

Let's LIST the program on paper to take a closer squint at it.

The line spacing stayed set where it was when we STOPped the program. That spacing is now the new "standard," replacing our power-up value of 12 dots = 1/6".

We can, however, easily shift back to the original (default) line spacing by just sending:

```
PRINT CHR$ (27) "2"
```

Try it, then type:

```
LIST
```

Ah! Sanity has returned. Whenever we send <ESC> "2," the spacing returns to the "power up" value of 6 lines per inch - 12 vertical dots per line.

There are two other ways of returning the line space to its original size:

1. Use the ESC A sequence we just learned:

```
>PRINT CHR$ (27) "A" CHR$ (12) , or
```

2. Turn the printer OFF, then ON.

Turn the printer OFF, then ON, before continuing to the next topic. Remember that this resets the Top of Form.

Caveat

Life is never as rosy as the advertisements. If it were, where would the challenge be?

There are several code numbers already reserved for special things. If we look closely at the printouts from this chapter, we can see that the spacing when the following numbers are used may not be what we have anticipated:

8 - reserved for (DELETE PREVIOUS CHARACTER)

9 - reserved for (TAB)

10 - reserved for Line Feed

12 - reserved for Form Feed

13 - reserved for Carriage Return

There are no nice clean ways around this problem -just very messy ones. (See Appendices D and F for ways to POKE these codes to the printer). Best we restrict our selection to the remaining numbers.

Underlining

Many computers do not have the capability of sending an ASCII character to temporarily suppress the Line Feed when a Carriage Return is sent. An Apple running BASIC with the Epson printer card falls in this category. We cannot do underlining by simply sending CHR\$ (13), a simple CR, then sending underlining dashes in the right places.

TRS-80 zealots can refer to Chapter 2, open up the printer, and move the auto line feed wire to the red terminal. This means absolutely no line feeds unless we send a CHR\$ (10) each time one is needed. This “technique” does permit underlining, but for all practical purposes, it’s impractical!

Underlining should not be confused with just moving down one line then printing a series of dashes. True underlining requires that we first print the line to which we want to add whole or partial underlining. We then Return the Carriage (print head), suppress the LF so the paper does not advance, and print the underline character ASCII 95. This is called “overstriking.” The underline character is printed in position 7 of our 12 x 6 character dot matrix. Users with LF suppression can also do such things as slashing zeros, sevens, etc.

There is another way to suppress the LF using the vertical spacing tricks we just learned. The method to be demonstrated is practical for such things as report headings or special emphasis, where the result is deemed worthy of the extra programming effort.

Type in this NEW program:

```
9 PR #1                               (Apple)
10 PRINT CHR$ (27) "A" CHR$ (0)       (or 0+ 128)
20 PRINT "UNDERLINE"
30 FOR U=1 TO 9 : PRINT CHR$ (95); : NEXT U
40 PRINT CHR$ (27) "2"
50 PRINT "WORKS"
59 PR #0                               (Apple)
```

and RUN.

Oops, too close for comfort. We forgot that the MX-70 doesn't have lower case descenders, and the underline character is in that category. We'd better drop the underline down one dot.

UNDERLINE
WORKS

Figure 4-6

Change line 10 to:

```
10 PRINT CHR$ (27) "A" CHR$ (1)
```

and RUN.

Compare this RUN with the previous one. It's looking better, isn't it? Let's drop the underline position down one more dot. Go ahead - can you figure out what to do?

Here is how the program works:

Line 10 sets line spacing to 1 DOT (1)

Line 20 PRINTs something we want to underline

Line 30 PRINTs 9 underlines. (Computers that support STRING\$ could use it here.) The ; suppresses the LF.

Line 40 returns line spacing to the normal 12 DOTS

Line 50 PRINTs more text to show that all is normal

and that's plenty to think about in this chapter.

Code Summary

CHR\$ (27) "A" CHR\$ (##) - SETS LINE SPACING TO ## DOTS
(0 <= ## <= 85)

CHR\$ (27) "2" - RETURNS LINE SPACING TO 12 DOTS

CHR\$ (10) - LINE FEED

CHR\$ (13) - CARRIAGE RETURN

CHR\$ (95) - UNDERLINE CHARACTER

Chapter 5

Graphtrax II

Caution - Entering High Resolution Space

To create graphics on the MX-70, we must enter a completely new mode: the GRAPHICS MODE. In this mode, pre-defined characters do not exist - only dots. We create our own characters or images by arranging the dots however we wish.

In the GRAPHICS mode we have complete control over which pins fire, and when. It sounds like a lot of fun, and is, but there are an awful lot of dots in just one line, let alone an entire page.

Think of it this way. There are 80 "regular" characters in a normal line (row). Each character is 6 dots wide (5 dots plus 1 blank column). That comes to 480 columns in each row. Good grief!

Fortunately, we don't have to fill all 480 columns on every line. In fact, the first thing we do when entering graphics mode is tell the printer how many columns of dots we will send it, a row at a time.



Warp 3

The graphics mode is entered by sending the printer a cluster or sequence of 3 codes:

FORMAT: `<ESC> "K" N1 N2`

EXAMPLE: `CHR$ (27) "K" CHR$ (15) CHR$ (0)`

`CHR$ (27)` is of course the Escape code, and Escape "K" means KICK into graphics mode.

`CHR$ (15)` says: "reserve 15 columns for graphics." `N1` can range from 0 to 255 (0-127 for 7 bit computers)

`CHR$ (0)` If `N2` is 0, it means "Ignore me, `N1` specifies the number of columns to expect."

If `N2` is 1, the number of columns will equal $N1 + 256$. If this sum of `N1` and `N2` exceeds 480, only 480 columns will be reserved.

The Saki flowed like melting snow from Mt. Fuji when they contrived that wild scheme. Let's see if we can wring it out.

Suppose we want to shoot 100 columns of graphics to the printer, on one row. What numbers would we use for N1 and N2?

CHR\$ (100) CHR\$ (0) for 100 columns

Ok, so we got lucky.

How about 300 columns? Well, $300-256=44$. Let's try:

CHR\$ (44) CHR\$ (1)

Since $N2 = \text{CHR\$ (1)}$, 256 is added to N1. $44 + 256 = 300$.

Not really so bad. Pass the Saki!

Enter these lines, but don't RUN:

```
9 PR #1          (Apple)  
10 PRINT CHR$ (27) "K" CHR$ (50) CHR$ (0);
```

TRS-80 Model I use:

```
10 LPRINT CHR$ (27) "K" CHR$ (50) CHR$ (2) ;
```

The Model I can't send CHR\$ (0) reliably, and ANY EVEN number is interpreted as a zero in the ESC K sequence here in Graphtrax II.

How many columns is line 10 reserving, 50 or 306?

Answer=50. Don't RUN yet.



Firing the Pins

There are 9 pins in the MX-70 print head. In graphics mode, we have control of the top 8 (the middle 7 for 7 bit computers).

We'll label these top 8 pins as follows:

128 - 0	TOP
64 - 0	
32 - 0	
16 - 0	
8 - 0	
4 - 0	
2 - 0	
1 - 0	BOTTOM
0	(Ninth pin not used)

From now on we will refer to the second pin (pin 1 above) as the “bottom” pin when using graphics.

Why not label them 1, 2, 3, . . . 8 etc? Well, the numbers shown are the actual ASCII numbers that fire the respective pins. CHR\$ (128) fires the top pin, while CHR\$ (1) k-es the bottom one. CHR\$ (7) fires the bottom three (4+2+1). Just no way to escape Binary math when dealing with computers, is there?

Add these lines:

```

20 FOR P = 1 TO 50
30 PRINT CHR$ (1);
40 NEXT P
50 PRINT
59 PR #0          (Apple)

```

and RUN.

Sure enough, fifty little dots. CHR\$ (1) in line 30 caused the bottom pin to fire. The semicolon suppressed the line feeds. The FOR-NEXT loop fired the bottom pin 50 times.

Change line 30 to:

```
30 PRINT CHR$ (127);
```

and RUN.



Figure 5-1

Wow! A jackpot. Using the bottom of an old Coke bottle as a magnifier, we see that each column is 7 dots high. 50 columns of 7 rows of dots. The sum of the bottom 7 pin numbers is

$$1 + 2 + 4 + 8 + 16 + 32 + 64 = 127$$

So that's how they do it!

Quiz Time

Question: What single number will allow us to fire only the pins labeled 1, 4, and 16? Think it through now. How about . . .

$$1 + 4 + 16 = 21 \quad ?$$

Let's try

```
30 PRINT CHR$(21);
```

and RUN.

Oh - winning is so much fun!

Don't be too concerned about what movie is playing on your video screen. It may not know how to handle these codes. It's what's happening on the printer that counts.

The Grand Scheme

This is starting to make sense. The "logic" pattern must be:

	0	1	2	3	4	5	6	7	8	9	10	11	12	-----	255
128															128
64															64
32															32
16															16
8										8
4											4
3					2
1				1
	0	1	2	3	4	5	6	7	8	9	10	11	12	-----	255

Graphic Codes

Now for the Bad News

Before racing off to create a HI-RES forgery of the Mona Lisa, be aware that the above is how it SUPPOSED to work. Reality is brutal - there are some exceptions to almost every rule.

Code numbers 9 and 13 for the Apple, and 0, 10, 11, and 12 for the TRS-80 create havoc when used as N1 or N2. (It's déjà vu from the last chapter). Other computers may have trouble with different codes. Even worse, the printer accepts code numbers from 0 to 255 to determine the dot pattern, but many computers are not able to send code numbers greater than 127. It's the old "missing eighth bit" caper. Like the "lost chord."

Sigh! Once again the printer can out perform a computer. In the Apple's case, the Epson Parallel Interface card deactivates the eighth bit so Apple users can control only 7 pins. If it didn't, bit 8 from the Apple would be on **all the time**, firing pin 8 every time. Most TRS-80s can control all 8 pins without difficulty.

Back to the Welcome Program

We've seen that each pin is associated with a number. The numbers 1, 2, 4, 8, 16, 32, 64, 128 all relate to the mathematical powers of 2. Binary math, and all that stuff. Here's the relationship:

$$2^7 = 128 \quad 2^3 = 8$$

$$2^6 = 64 \quad 2^2 = 4$$

$$2^5 = 32 \quad 2^1 = 2$$

$$2^4 = 16 \quad 2^0 = 1$$

Let's see if we can fire the pins, in sequence, from the bottom up. Make the program read:

```

9 PR #1                                (Apple)
10 PRINT CHR$ (27) "K" CHR$ (50) CHR$ (0) ;
20 FOR P = 0 TO 6
30 PRINT CHR$ (2^P) ;
40 NEXT P
50 GOT0 20

```

TRS-80 Model I:

```
10 LPRINT CHR$ (27) "K" CHR$ (50) CHR$ (2);
```

and RUN.

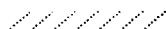


Figure 5-2

Press CTRL-C or BREAK to stop. Very impressive.

Note that we are not printing the slash character (ASCII number 47); but are positioning the dots one at a time. Line 50 of the “welcome program” should now make sense. We see that to fire a single pin, we must send a power of 2 down the line to the printer. Line 50 of the “welcome program” does just that.

```
50 PRINT CHR$ (2^INT(3.4*SIN(I)+1)));
```

The entire expression to the right of the exponent sign chooses an appropriate exponent for 2. The exponents determine which pin is fired, giving us a nice sinusoidal curve. Math types love it.

Who Sez We Ain't Got No Class?

Let's put the printer through some “aerobic exercises” with this NEW program:

```
9 PR #1 (Apple)
10 PRINT CHR$ (27) "A" CHR$ (7)
20 FOR R=1 TO 3
30 PRINT TAB (10);
40 PRINT CHR$ (27) "K" CHR$ (14) CHR$ (0);
50 FOR N=1 TO 14
60 READ D : PRINT CHR$ (D);
70 NEXT N : PRINT : NEXT R
80 PRINT CHR$(27) "2"
99 PR #0 (Apple)
```

```

100 DATA 4,4,4,4,4,61,71,70,60,4,4,4,4,4
110 DATA 72,79,89,107,77,73,127,89,73,79,73,73,73,73
120 DATA 16,112,54,17,81,17,126,16,48,80,16,16,16,16

```

TRS-80 Model I:

```
40 LPRINT CHR$ (27) "K"CHR$ (14) CHR$ (2) ;
```

and RUN.



Figure 5-3

How about that one, culture hounds? Here's how it works:

Line 10 sets the line spacing for 7 pin graphics. The G clef is printed in three lines.

Line 20 takes care of the looping.

Line 30 moves the figure away from the left margin.

Line 40 kicks the printer into graphics mode - 14 columns per line.

Lines 50-70 read each code from the DATA lines and send it to the printer.

The NEXT R in line 70 sends control back to line 20 to start the next line.

Line 80 returns the line spacing back to 6 lines per inch.

How's that for a nifty tune?

Code Summary

ESC K N1 N2 - ENTERS GRAPHICS MODE. N1 and N2 DETERMINE LENGTH OF GRAPHIC LINE.

0 <= N1 <= 255
0 <= N2 <= 2

Chapter 6

Advanced Graphics

Firing Pins at Seven Paces

The MX-70 allows any computer to control the firing of its 8 active pins by simply sending ASCII code numbers between 0-255. Conversely, these 256 code numbers are sufficient to control all 8 pins. Multi-line graphics programs will therefore usually be done with a top-of-line to top-of-line spacing of 8 dots (8/72").

As already repeated, many computers, or their printer interface boards, will not control or pass code numbers greater than 127. These computers can control only 7 of the 8 active pins in graphics mode. With that common computer deficiency in mind, the examples in the remaining chapters are limited to 7 dots per line.

The limitation is not a crippling one. We just have to adjust for it. (The same field of corn can be harvested by a 2-row corn picker as by a 4-row - it just takes a little longer.)

Users with 'superior' computers can follow right along at our 7 dot clip, knowing that their computers can drive all 8 pins if desired, but not until we've all finished the course.

Housekeeping

Let's develop a fairly complex graphics picture a step at a time. Everybody start by typing in these lines:

```
9 PR #1                      (Apple)
10 PRINT CHR$ (27) "A" CHR$(7)
20 N=50
30 PRINT CHR$ (27) "K" CHR$ (N) CHR$ (0);
```

TRS-80 Model I:

```
30 LPRINT CHR$ (27) "K" CHR$ (N) CHR$ (2);
```

but don't RUN yet.

Line 10 sets the top-to-top line spacing at 7 dots.

Lines 20 and 30 KICK the printer into graphics mode. N specifies the number of graphic columns. By making it a variable, we can easily change it later, right within a program, to print graphic lines of different lengths.

As a preliminary software test before things get too complicated, we'd better check the line spacing. Add these lines:

```
25 FOR X=1 TO 3
40 FOR C = 1 TO 50 : PRINT CHR$ (127); : NEXT C
501 PRINT : NEXT X
```

... and we should always return the line spacing to its "normal" power-up figure of 12 by ending every program with:

```
900 PRINT CHR$ (27) "2"
999 PR #0 (Non-Apple owners use END instead)
```

Now we can RUN.

After 3 passes of the print head, our printout should look like this:



Figure 6-1

What Are We Trying To Do?

We are creating a universal program that will READ and process large amounts of DATA, and is simple to use. READ and DATA statements are used since the DATA need be entered in the program only once, and DATA lines are easily edited. Think about this paragraph and understand it since the going will get rougher before it gets easier.

Let's delete lines 25 and 50 and change line 20 so it will READ in a value of N, the number of dots to be printed in a specific row:

```
20 READ N : PRINT TAB(10);
```

The Data Bank

HI RESolution graphics requires lots of DATA. It comes with the territory

So, what should our DATA lines look like? How should we format them so they contain the information we need? What information do we need?

ASCII numbers 0-127 are used to specify the firing combinations for the 7 pins. Since much graphic printing, fortunately, is repetitious, we can develop a coding scheme to simplify and reduce the sheer volume of DATA required. It can use pairs of numbers.

To avoid needless typing, let's use a minus sign as a "flag" for the first of the number pair to indicate that its value tells the number of times something is to be repeated. The second value of the pair will specify the desired combination of pins to be fired. For example:

```
100 DATA -42,127
```

means: "fire all 7 pins (#127) 42 times." Think that one through before continuing.

Got it?

We also have to tell the printer the number of columns to print on each line. Let's dedicate the first number in each DATA line to that purpose.

Enter this line of DATA:

```
1020 DATA 55,0,3,7,15,31,63,63,-42,127,63,63,31,15,7,3
```

Recheck the numbers carefully to be sure they are copied correctly.

The first number tells the MX-70 to expect 55 bytes of graphic information. The rest of the numbers specify the pin combinations to be fired in each column, except for that suspicious number pair in the center: -42,127. Remember what it means?

The Program

We also need a "DO LOOP" that will READ the DATA:

```
40 FOR G=1 TO N
50 READ X
120 NEXT G
130 PRINT
```

The numbers READ into X are the actual pin firing instructions. If X falls in the range 0 - 127, we print it in line 60 below. It also includes a “filter” to snag any negative numbers:

```
60 IF X>=0 THEN PRINT CHR$ (X); : GOT0 120
```

TRS-80 Model I:

```
60 IF X>=0 POKE 14312,X : GOT0 120
65 IF PEEK (14312) <> 63 THEN 65
```

If X is negative, the line 60 test fails and execution falls through to:

Line 70, below, where the “G” FOR/NEXT loop is reset and the next value READ into Y from the DATA line.

Line 100 PRINTs it ABS (X) times:

Add these lines:

```
70 G = G-X-1
80 READ Y
90 FOR P = 1 TO ABS (X)
100 PRINT CHR$ (Y);
110 NEXT P
```

TRS Model I:

```
100 POKE 14312,Y
105 IF PEEK(14312)<>63 THEN 105
```

List and recheck the completed program to make sure everything is correct:

```

9 PR #1                      (Apple only)
10 PRINT CHR$(27) "A" CHR$(7)
20 READ N : PRINT TAB (10);
30 PRINT CHR$(27) "K" CHR$(N) CHR$(0);
40 FOR G=1 TO N
50 READ X
60 IF X>=0 THEN PRINT CHR$(X); : GOTO 120
70 G = G-X-1
80 READ Y
90 FOR P = 1 TO ABS(X)
100 PRINT CHR$(Y);
110 NEXT P
120 NEXT G
130 PRINT
900 PRINT CHR$(27)"2"
999 PR #0                      (Non-Apple use END)
1020 DATA 55,0,3,7,15,31,63,63,-42,127,63,63,31,15,7,3

```

Summary of TRS-80 Model I changes:

```

30 LPRINT CHR$(27)"K"CHR$(N)CHR$(2);
60 POKE 14312,X
65 IF PEEK(14312)<>63 THEN 65
100 POKE 14312,Y
105 IF PEEK(14312)<>63 THEN 105

```

Finally, we are ready to try it out . . . so RUN.



Figure 6-2

Big deal! Maybe if we water it, it will grow.



All That Work for one Line?

We shouldn't complain too loudly. It works, and no one said creating graphics was easy.

Our current “universal” program can only handle one line of DATA, and it’s becoming increasingly apparent that HI-RES graphics requires lots of lines of DATA. The following simple changes allow it to handle virtually unlimited amounts of DATA:

```
15 ONERR GOTO 900      (TRS-80 etc. use on ERROR GOTO)  
130 PRINT : GOTO 20
```

Line 130 creates an infinite loop, sending execution back to the READ statement after printing each line.

Line 15 gives the program a smooth ENDing when the DATA is all READ. When there is no more DATA, an OUT-OF-DATA error message appears, and the program is cleanly ended, even resetting the printer line spacing back to its “power-up” default.

Let’s add 2 more lines of DATA to see how it works:

```
1000 DATA 49,-8,0,3,15,63,-34,0,63,15,3,0  
1010 DATA 49,-7,0,63,-3,127,7,-32,0,7,-3,127,63
```

and RUN.



Figure 6-3

It sprouted flowers!

Rowdy Characters

Add the following lines:

```
1070 DATA 44,-12,8,-32,127
1080 DATA 39,-17,0,64,96,112,120,124,-12,127,124,
      120,112,96,64
```

and RUN.



Figure 6-4

Oops! There's trouble in River City. Fortunately, we've seen these same control code problems before. Hate to keep bringing this subject up, but it does disturb our use of the printer.

We've seen codes 9 and 13 cause trouble on the Apple, and codes 0, 10, 11, and 12 cause trouble on the TRS-80. If we send them to the printer via CHR\$, it interprets them as control codes instead of pin firing instructions. Unfortunately, that makes sense.

Code 8 can also cause trouble since it means "delete the previous character sent to the printer and stored in its print buffer." In line 1070 we tried to send a whole sequence of them, and on some computers the program blew up!

For Experts Only

The errors caused by these codes do not always occur, but it is best to avoid them if possible. Short of rewriting the program to circumvent these numbers, one way is to bypass the PRINT statement and POKE our graphic codes directly to the printer driver in the computer's memory. That memory address is 49296 (CO90 Hex) in the Apple, and 14312 (37E8 Hex) in the TRS-80 Model I (See appendices for examples of use). Other computer users can check their computer's technical manual for its printer driver memory address.

The advantage of POKE over PRINT CHR\$ in this specific situation is that it bypasses some of the nasty tricks we keep encountering with BASIC. The disadvantage is that indiscriminate POKEing can cause unbelievable software crashes. Unless you really understand POKEs, best to live with the limitations inherent with your computer/printer combination, and avoid troublesome codes when possible. Model I users will need to use the POKEs shown for this particular example.

A Confession

Line 1070 was artificially injected into this conversation to illustrate some of the problems the rebel character codes can cause. We're playing for real now, so change line 1070 to:

```
1070 DATA 44,-12,0,64,112,120,124,127,127,15,7,67,99,51,3,  
       63,123,71,3,3,71,123,63,3,51,99,67,7,15,127,127,  
       124,120,112,64
```

and RUN.



Figure 6-5

Ah SO! Is it a picture of the MX-70 design engineer? Turned up-side down it looks like a flying saucer.

To see the final picture we have to type a few more lines. (Think of it as the price of greatness.)

```
1030 DATA 56, -11,127,123,124,126,-28,127,126,124,123,  
       -11,127  
1040 DATA 55,0,96,120,127,126,-7,127,63,15,3,125,112,120,  
       124,126,-16,127,126,124,120,112,125,3,15,63,  
       -7,127,126,127,120,96  
1050 DATA 52, -4,0,112,60,94,111,119,119,123,123,-32,127,  
       123,123,119,119,111,94,60,112  
1060 DATA 49,-7,0,64,112,120,124,-7,127,120,119,120,115,  
       116,120,112,112,113,112,112,113,112,112,120,116,  
       115,120,119,120,-7,127,124,120,112,64
```

Whew! Lots of DATA. The price of glory is high, but this is ridiculous.

Cross your fingers and RUN.



Figure 6-6

It was worth it!

We now have all the tools needed to design and print our own graphics. Be sure to save the finished program as we will use it in the next chapter.

Code Summary

49296 - POKE LOCATION FOR APPLE TO SEND INFO TO PRINTER

14312 - POKE LOCATION FOR TRS-80 MODEL I

Chapter 7

The Final Push

Only masochists failed to SAVE the final program from the last chapter. We're now going to delete all its DATA lines and reconstruct it for more universal applications.

DELETE LINES 1000-1080.

Mix and Match

We are obviously amazed at the MX-70's graphics ability. Our thoughts logically lead to thinking about mixing graphics and text on the same line. Can it be done? Sure, nothing to it. Just be sure to hang trailing semicolons on the ends of lines as needed to keep the printer from doing unwanted Line Feeds.

A few swift chops to our existing program will demonstrate the point. We can use it as a SUBroutine to READ and PRINT a graphics string.

Change the following:

```
10 PRINT CHR$ (27) "A" CHR$ (l0)
15 GOTO 140
130 RETURN
```

Line 10 increases the line spacing to give our next figure a little breathing room (10 dot line height.)

Lines 15 and 130 turn the first part of the program into a subroutine.

Let's start with a single line of graphics to set the stage:

```
140 GOSUB 20 : PRINT
150 DATA 60, -27, 127, 126, 100, 64, 1, 11, 63, -27, 127
```

and RUN.



Figure 7-1

OK. The subroutine works.

No, it's not another demon. It is a "broken signboard" that needs a text type LABEL in the middle. The finished graphics/text display will be printed in these three parts:

15 columns of graphics
30 columns of text (5 characters)
15 columns of graphics

Now we have to mix and match graphics and text on the same line. First, the graphics:

```
160 GOSUB 20
170 DATA 15,127,8,28,62,93,-8,28,0,0
```

and RUN.



Figure 7-2

Then text:

```
180 PRINT"82 CM";
```

and RUN.



Figure 7-3

And more graphics:

```
190 GOSUB 20
200 DATA 15,0,0,-8,28,93,62,28,8,127
```

and RUN.

```
*****.*****
P---82 CM ---H
```

Figure 7-4

Nothing to it!

NOTE: As long as semicolons are used to delay the line feeds, graphics and text are easily mixed on the same line. New lines can be started by simply adding line feeds with PRINT statements as shown in line 140.

The Long Lines Division

None of our examples so far have used the full 480 columns available to the MX-70. In fact, we've always used less than 100 columns since the printer will be used by computers with different capabilities. As we expand the widths of our displays, these differences quickly become more obvious.

TRS-80 type computers, for example, can send a full range of ASCII codes 0 to 255 (8 bits) via the CHR\$ function. (A few must be poked, as we've seen.) These computers can therefore choose any graphics width from 0 to 480 columns.

Printer cards for Apple type computers typically pass ASCII numbers from 0 to 127 (7 bits). This forces us into extra programming and devious means to utilize the MX-70's complete range of 480 graphics columns.

In any one <ESC>“K” sequence, 7-bit computers are restricted to the ranges 0-127 and 256-383. Seeing is believing, so:

DELETE lines 150-200, and add:

```
150 DATA 150,-150,127
```

and RUN.

```
*****
```

Figure 7-5

Here's what happened. The 150 cannot be passed, so is reduced by 128, and sent to the printer as 22. Apple users see a bar 22 columns long (did you count them?). Users with 8-bit computers see a bar 150 columns long.



If we really understand what we're doing, the 7-bit limitation can be overcome. Apple users try:

```
140 GOSUB 20 : GOSUB 20 : PRINT
150 DATA 23,-23,127,127,-127,127
```

and RUN.

Now we are getting somewhere. Instead of trying to print the entire line in one shot, we hooked two print routines together with a semicolon. The first printed 23 columns, and the second printed 127 columns.

Really Long Lines

Suppose we want to use even longer lines, up to the maximum of 480 columns. To specify a line length greater than 255, the last number in the <ESC>"K" N1 N2 sequence must be a 1. Instead of changing line 30, we can accommodate the entire range of line widths by adding:

```
25 IF N>255 THEN PRINT CHR$ (27) "K" CHR$ (N-256) CHR$ (1);
: GOTO 40
```

If the desired line width (N) is greater than 255, CHR\$ (1) adds 256 columns, and CHR\$ (N-256) takes care of the difference. Let's test this upgrade by changing:

```
10 PRINT CHR$ (27) "A" CHR$ (7)
140 GOSUB 20 : PRINT
150 DATA 300,-300,127
```

and RUN.



Figure 7-6

We don't have to count the dots to check it. There are 60 dots per inch, so the line should be 5 inches long.

Apple Stew

I'm afraid that Apple users wound up with a slightly different result, even though the 300 should be valid for the current program.

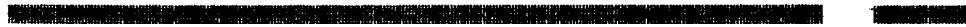


Figure 7-7

This appears to be a conflict among the Apple, the interface board, and the MX-70. It seems that anything printed past column 256 gives us scrambled eggs. While the factory-type experts work on a solution, Apple users add the following line:

7 PRINT CHR\$(9); "255H" (Apple users only)

and RUN.

Why it works is still a mystery to those that found this solution, but it does work! Use it whenever your Apple programs require continuous graphic strings of more than 255 columns.

Unfortunately, the biggest number of columns 7-bit computers can send is 383, the sum of $256 + 127$. We can print 256 columns by making N2 equal to 1, plus the maximum of 127 more. If we try and print 384 columns, that extra 128 is intercepted, reduced to zero, and we end up with 256. If these details interest you, play with line 150.

Apple Pie

BUT fear not! We couldn't end this happy tale on a sour note. Epson has provided a far superior way for Apple users to create high resolution graphics in BASIC. A "screen dump" program for use with Apple disk systems completely eliminates the need for wrestling with the above problems. Apple disk system users GOT0 Chapter 8 RIGHT NOW for the solution, and omit the rest of this chapter.

Apple non-disk users continue on with the rest of us.

Pictogram

Let's see if we can parlay strings of graphic characters into a useful format. Enter this NEW program.

```
9 PR #1                                (Apple)
30 FOR I=1 TO 5 : READ A(1) : NEXT I
40 DATA 8,107,92,107,8
60 FOR R=1 TO 5
80 READ N
90 FOR I=1 TO N : GOSUB 300
100 NEXT I : PRINT
120 NEXT R
130 DATA 30,25,17,11,8
140 PRINT CHR$(27) "2"
198 PR #1                                (Apple)
199 END
300 PRINT " "CHR$(27) "K" CHR$(S) CHR$(B);
310 FOR C=1 TO 5
320 PRINT CHR$(A(C));
330 NEXT C : RETURN
```

TRS-80 Model I:

```
300 LPRINT " "CHR$(27) "K" CHR$(S) CHR$(2);
```

and RUN.

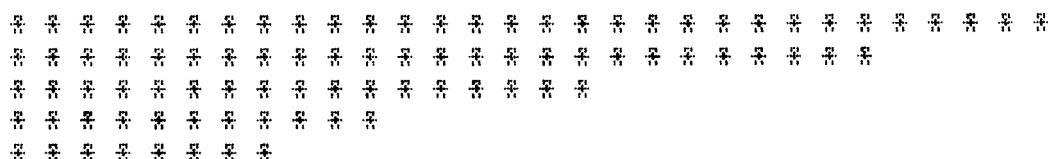
A 10x10 grid of asterisks (*). The grid is arranged in a specific pattern to form the outline of an apple. The top row has 10 asterisks. The second row has 9 asterisks. The third row has 8 asterisks. The fourth row has 7 asterisks. The fifth row has 6 asterisks. The sixth row has 5 asterisks. The seventh row has 4 asterisks. The eighth row has 3 asterisks. The ninth row has 2 asterisks. The tenth row has 1 asterisk. This creates a central vertical column of 10 asterisks, with the number of asterisks decreasing as you move outwards to the left and right.

Figure 7-8

The DATA for the population figure is READ into array A in line 30.

Each figure is printed in SUBroutine 300.

DATA line 130 determines how many figures are printed on each row.

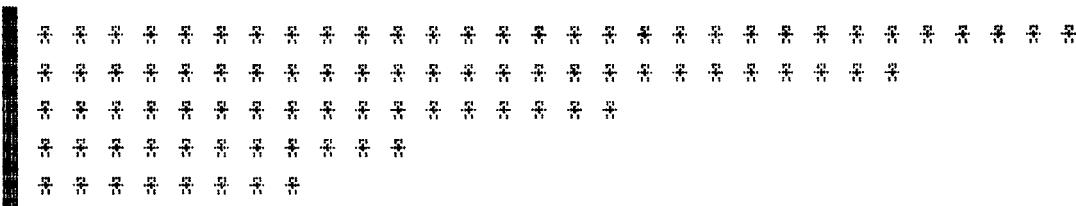
We have the makings of a horizontal bar graph, but it needs some trimmings. Let's add a few labels and move the whole works over a bit.

```
20 PRINT : PRINTCHR$(27)"A"CHR$(7)
50 GOSUB 200 : PRINT
70 PRINT TAB(5)1990-R;
80 GOSUB 200 : READ N
110 GOSUB 200 : PRINT
200 PRINT TAB(12)CHR$(27)"K"CHR$(5)CHR$(PI);
210 FOR I=1 TO 5 : PRINT CHR$(127); : NEXT I
220 RETURN
```

TRS-80 Model I:

```
200 LPRINT TAB(12)CHR$(27)"K"CHR$(5)CHR$(2);
```

and RUN.



Year	Population (approx.)
1985	10
1986	15
1987	20
1988	80
1989	80

Figure 7-9

That's better. Subroutine 200 prints the vertical bar that separates the labels and the graph.

If we add a quick title and legend, we'll be done.

```
10 PRINT TAB(23)CHR$(14)"POPULATION PROJECTION"  
140 PRINT : PRINT TAB(37); : GOSUB 300  
150 PRINT "( = 1,000"
```

and RUN.

POPULATION PROJECTION

Figure 7-10

That's plenty for this chapter.

Code Summary

PRINT CHR\$ (9) "255H" - Allows Apple users to print past column 255 without heartburn.

Chapter 8

Using the HIRES Screen Dump Program

Should I Study This Chapter?

Epson dealers can supply a machine language “screen dump” diskette for Apple users with 48K disk systems. It permits creation of exotic HI-RES graphics on the Apple screen, then dumping them, dot for dot to the MX-70 printer.

Since the MX-70 graphics are related to the Apple (as the MX-80 graphics replicate the TRS-80's), this particular software is of interest only to Apple users. Knowing this crazy field however, some ingenious software writer working in his garage will probably come up with a partial HIRES program for use with our popular computers.

Meanwhile, if you are using your MX-70 with another brand of computer, you can just as well ignore this chapter. Go in peace, and enjoy your printer.

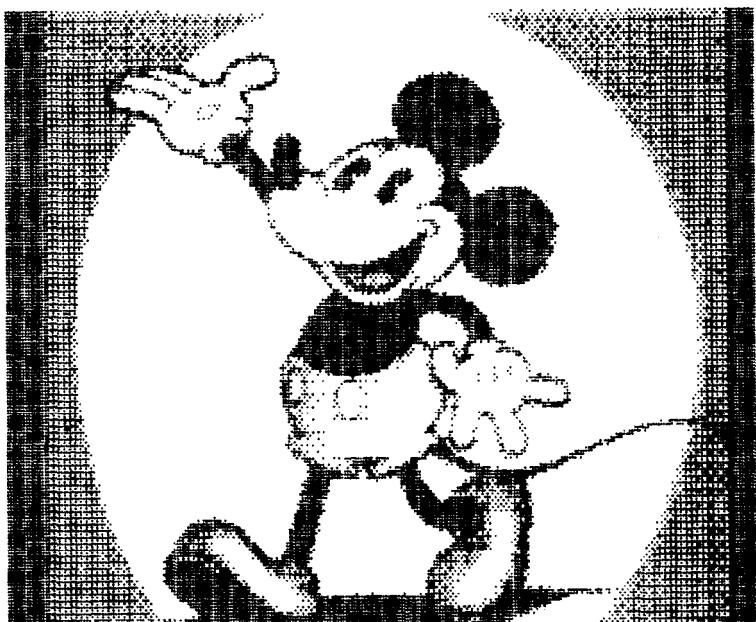


Figure 8-1

Apple Dump

The “screen dump” diskette comes with a DOS 3.3 operating system, so it works fine on a one drive system.

Load the demo disk into drive 1 and boot the system. The HELLO program will automatically load and execute. After a brief commercial from the copyright holder, the program asks:

IS THE PICTURE ALREADY IN MEMORY? (Y/N)?

We respond with:

N <RETURN>

The computer retaliates with:

INSERT DISK CONTAINING PICTURE
THEN PRESS RETURN
?

Since the DOS and sample PICTUREs are on the same disk, press

<RETURN>

What follows is the HIRES diskette catalog of sample programs and files. The CATALOG includes these programs:

HELLO

HIRES INSTRUCTIONS

EPSON.HIRES.OBJ

The HELLO program loads the screen dump routine and prompts us with questions. This is the program we are running now.

The HIRES INSTRUCTIONS program gives hints on how to use the screen dump.

The EPSON.HIRES.OBJ program is the actual screen dump routine written in machine code. It can be accessed with the USR command from BASIC.

The rest of the tiles are HI-RES demonstration pictures, and at this writing include:

BESSEL 2

BESSEL

INVADERS

EPSON DEMO PIC

CHART

DISNEY CHAR

SCHEMATIC

PORTRAIT

CHESS

Follow the Yellow Brick Road

The HELLO program guides us through the actual process of printing the pictures. It's completely self-explanatory (as every confusing technical book says), but here's some step-by-step guidance.

WHAT IS THE FILENAME OF THE PICTURE
?

The program is asking for the NAME of the picture we want to print. Let's try:

PORTRAIT

The computer reports:

LOADING PICTURE

then

IS THIS THE CORRECT PICTURE (Y/N);

?

flashes before our very eyes. We type:

Y <RETURN>

It wants to know more:

WHAT SIZE? (R=REGULAR L=LARGE)?

Oh, regular size is fine:

R <RETURN>

Still more options, so the next query is:

NORMAL(N) OR REVERSE(R)?

This can be a bit confusing. NORMAL prints every white dot on the screen as a black dot on the white paper. This IS what we "normally" think of as "normal" printing. If we want the picture to appear with a black background, exactly as it appears on the screen, we request REVERSE. Got that?

Chapter 8

For now, let's try REVERSE:

R <RETURN>

The last question is:

WHAT IS THE PRINTER SLOT # ? (1-7)
?

Double check to see that the printer is ON and loaded with paper. Our printer card is in slot I, so:

1 <RETURN>

Now we just sit back and relax while the MX-70 does its thing. No fair peeking until it finishes.

All together now-PEEK!



Figure 8-2

Wow, how long do you think it took the artist to create that picture? Can you imagine arranging all those little dots the way we learned in the last few chapters?

There are on the market cameras and digitizers that will convert pictures to electronic impulses and place them in the computer's memory. We can create pictures in memory by whatever means are available, then use the Epson HIRES program to dump them to the printer.

Go ahead and try the different print options with PORTRAIT, large and small, reverse and regular.

Print several of the pictures. Instead of PORTRAIT type the name of another one. Most load into the HIRES page "low memory." We'll wait for you here.

Be an Artist for Fun-and Profit

Impressed, or just intimidated?

Let's see what we can do' about making our own mess on the paper. Return to BASIC and type:

```
PR #1
RUN HIRES INSTRUCTIONS
```

(Don't worry if the screen says Epson TX-80 HIRES GRAPHIC ROUTINE - it's been modified for the MX-70.) Abort the program, then enter:

```
PR #0
```

Tear off this list of instructions so we can refer to them as we scurry along. According to these instructions, we can write our own program to fill a memory page with HIRES graphics, then use the USR function to print it out. Let's see if they're just putting us on.

Type in this short program:

```
10 HOME
20 HGR
30 HCOLOR=7
40 FOR A=0 TO 20.4 STEP .02
50 R=10 * A * COS(A) * SIN(A)
60 HPLOT 138 + R * COS(A) , 79 + R * SIN(A)
70 NEXT A
```

and RUN.



Not too shabby for rank amateurs.

Here's what happens:

Line 10 clears the screen.

Line 20 selects the “low memory” HIRES page. This part of memory is used for high-resolution graphics, while text is stored on a different ‘page’ of memory.

48K disk users have two different chunks of memory that can be used for HIRES graphics. If we wanted, we could store 2 different pictures at the same time. Refer to the Apple Programming Reference Manual for more information.

Line 30 selects a color.

Lines 40 and 70 set up a loop for plotting a figure.

Line 50 calculates the radius as a function of the angle. We are using polar coordinates in case you hadn't guessed. If you don't really care, it doesn't really matter - the picture is still pretty.

Line 60 converts from polar to rectangular coordinates and plots a single point. Whew! It's a good thing computers don't require a good math background.

Why All This Math?

Using Math equations to create graphics is easier than by doing it the hard way, point-by-point. They do, however, restrict us to graphs or plots of natural phenomena, and who among us will claim credit for creativity in that area?

Type:

TEXT

to return back to the TEXT page. Even though we can't see it, the picture remains stored in the HI-RES page of memory.

We can play with line 50 to get different figures. Try:

50 R=3 * A and

50 R=30 * (~ - SIN(A)) and

50 R=80 * COS(3 * A/4)
(Change line 40 to 40 FOR A=0 TO 26 STEP .01)

and RUN.

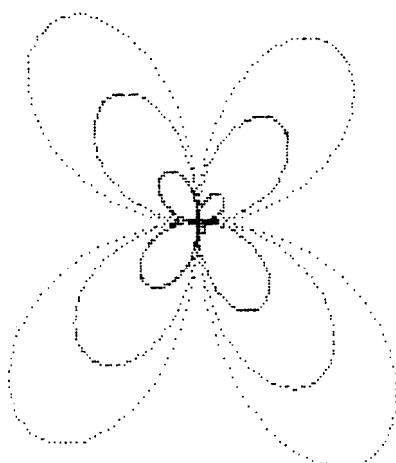


Figure 8-3

When you see one you like, add this screen dump “linking routine” to hook the BASIC program to the HIRES program:

```
80 POKE 10,76 : POKE 11,00 : POKE 12,96
90 PRINT "BLOAD EPSON.HIRES.OBJ"
100 PRINT USR(0001)
```

and RUN.

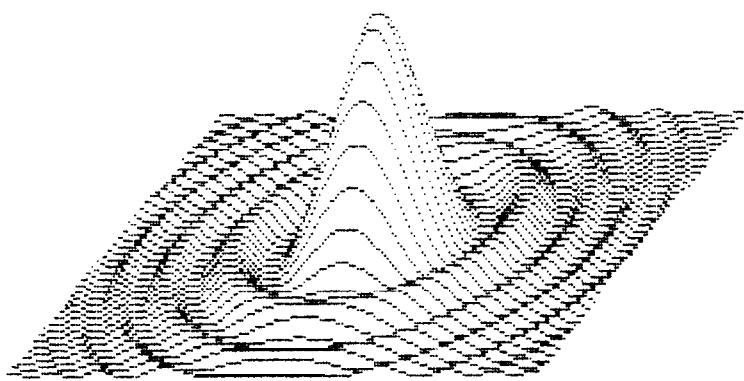


Figure 8-4

Lines 80 and 90 are straight out of the HIRES INSTRUCTIONS program listing. They load the screen dump routine without leaving the BASIC program.

As indicated in the HIRES instructions, we use the machine language routine named EPSON.HIRES.OBJ (Line 90), and

where:

- A = PAGE (O=PG.1, l=PG.2)
- B = SIZE (O=SMALL, l=LARGE)
- C = PICTURE (O=REG., l=INVERSE)
- D = PRINTER SLOT (1 TO 7)

Therefore, our line 100 calls the routine to print the picture with:

Page 1 (Low) of memory,
Small size,
Regular picture, and
Printer slot # 1.

Change the USR statement in line 100 to suit your printing desires.



“...Never give up!”

Commencement Address

Now that we've learned everything there is to know about the MX-70 and HIRES graphics, our education is ready to begin. I'd enjoy hearing from you and seeing some of your artistic creations.

Appendix A

ASCII Charts for MX-70

		SP	Ø	@	P	l	p
0	16	32	48	64	80	96	112
1	17	!	1	A	Q	a	q
2	18	*	2	B	R	b	r
3	19	#	3	C	S	c	s
4	20	DC4	\$	4	T	d	t
5	21	%	5	E	U	e	u
6	22	&	6	F	V	f	v
7	23	'	7	G	W	g	w
BS		(8	H	X	h	x
8	24)	9	I	Y	i	y
9	25	*	:	J	Z	j	z
LF		:	;	K	[k	i
10	26	42	58	74	90	106	122
ESC		+	;	75	91	107	123
FF		.	<	L	92	108	124
CR		-	=	M	93	109	125
SO		..	>	N	94	110	126
15	31	/	?	O	95	111	127

Appendix A

This is the standard ASCII character set. The codes that are highlighted are understood by the MX-70 printer. The rest are for your information only, and serve no function on the MX-70 printer.

		DEC	0	16	32	48	64	80	96	112
	HEX.		0	1	2	3	4	5	6	7
DEC	HEX.	BCD	0000	0001	0010	0011	0100	0101	0110	0111
0	0	0000	NULL	DLE	SP	0	(@	P		p
1	1	0001	SOH	DC1	!	1	A	Q	a	q
2	2	0010	STX	DC2	“	2	B	R	b	r
3	3	0011	ETX	DC3	#	3	C	S	c	s
4	4	0100	ET	DC4	\$	4	D	T	d	t
5	5	0101	ENQ	NAK	%	5	E	U	e	u
6	6	0110	ACK	SYN	&	6	F	V	f	v
7	7	0111	BEL	ETB	‘	7	G	W	g	w
8	8	1000	BS	CAN	(8	H	X	h	x
9	9	1001	HT	EM)	9	I	Y	i	y
10	A	1010	LF	SOB	*	:	J	Z	j	z
11	B	1011	VT	ESC	+	;	K	[k	{
12	C	1100	FF	FS	,	<	L	\	l	-
13	D	1101	CR	GS	-	=	M]	m	}
14	E	1110	SO	RS	.	>	N	^	n	~
15	F	1111	SI	US	/	?	O	-	o	

Appendix B

Control Codes

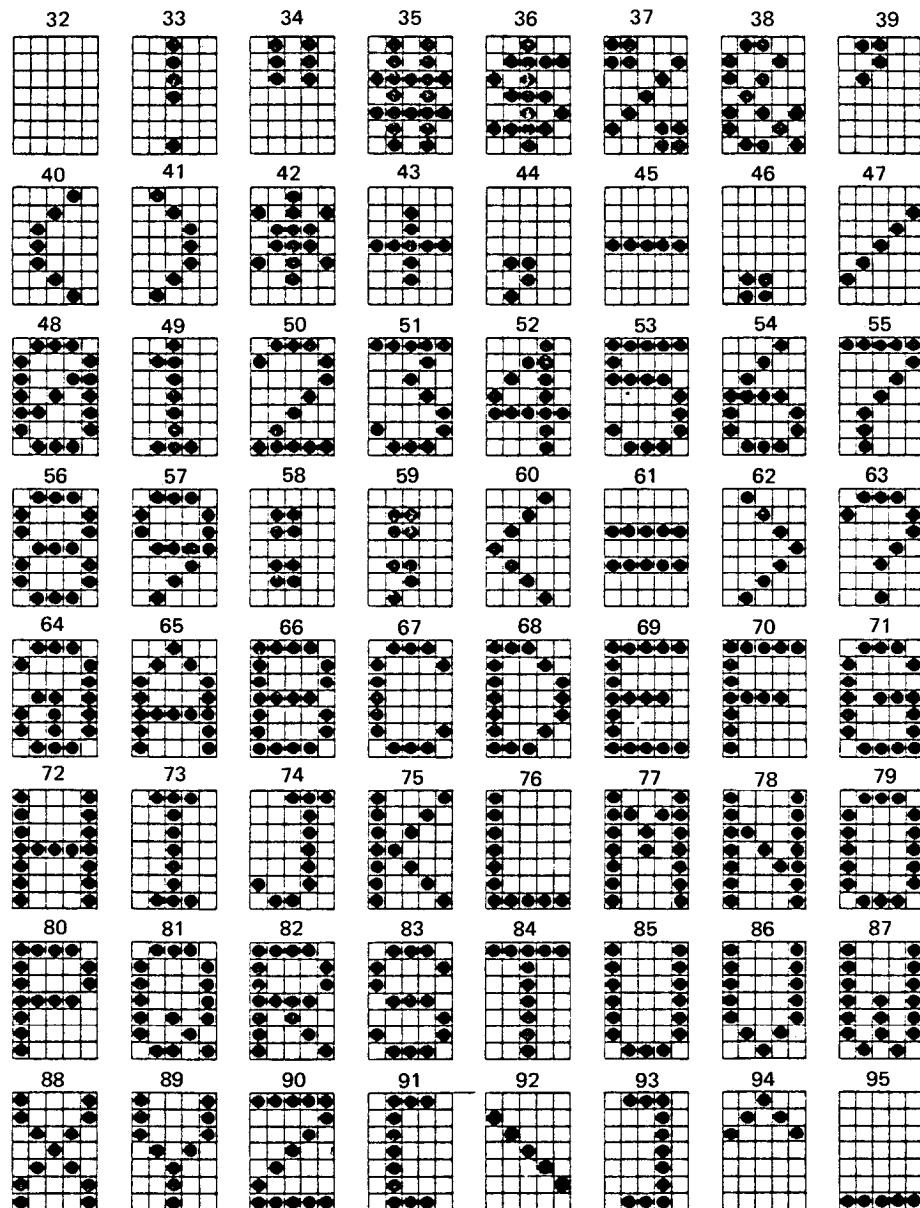
ASCII	FUNCTION
*2	Returns line spacing to 1/6".
8	Cancels last character entered in text mode.
10	Line Feed (LF). Empties printer buffer.
%12	Form Feed (FF). Advances paper to the next top of form. Empties printer buffer.
13	Carriage Return (CR). Empties printer buffer.
14	Turns on double width.
20	Turns off double width.
27	ASCII code for ESCAPE.
*:65	ASCII "A." Sets line spacing of line feed. Range: 0 - 85/72".
*67	ASCII "C." Sets form length. Default is 66 lines. Range: 1-127 lines.
*75	ASCII "K." Initiates graphics mode.

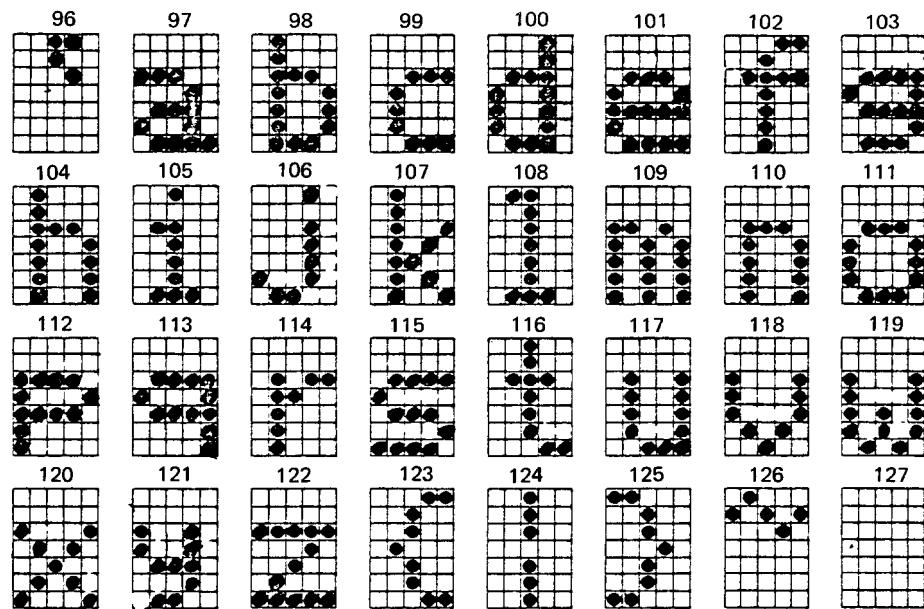
NOTE: Numbers flagged with a % **may** require the addition of 128 to make them work reliably on the TRS-80.

Numbers flagged with a * **must** be preceded by the ESCAPE code CHR\$ (27).

Appendix C

Character Set





Appendix D

TRS-80 Differences

This MX-70 Manual is designed around the Apple II for reasons discussed in Chapter 2. Fortunately, TRS-80 users can use it successfully by following the notes and suggestions in this appendix and throughout the text. In fact, using the MX-70 with the TRS-80 is quite easy: with no ports to initialize, no control codes needed to permit 80 column listings, and straightforward LPRINT commands.

General Changes

All TRS-80 models use LPRINT and LLIST to send information to the printer. Be sure to change all PRINTs intended for the printer to LPRINT.

Ignore all references to PR#1 and PR#0. They are for the Apple, as noted.

It may be necessary (and is usually permissible) to add 128 to certain code numbers, (but not when in graphics mode). For example, if CHR\$(27) "A" CHR\$(12) doesn't work, use CHR\$(27) "A" CHR\$(140). The problem is with the computer.

TRS-80 Model I Without Expansion Interface

Model I TRS-80 owners may use Radio Shack printer interface #26-1411 in place of the Radio Shack expansion interface, if necessary.

TRS-80 Model I With Expansion Interface

Graphics Codes

Sending graphics codes to the MX-70 via Model I BASIC has its drawbacks. Entering graphics mode can be an adventure if we try, for example, to specify a line length of 10. BASIC intercepts this code, and passes it down the line as a 13. So the printer expects 13 columns though we sent it only 10.

Once in graphics mode, certain codes must be sent via POKE rather than LPRINT CHR\$. The Model I printer driver interprets CHR\$(0) as a null code and does not pass it on to the printer. That hurts since we need this code in graphics mode.

We can get around this limitation by POKEing the code numbers directly to the printer buffer, foiling BASIC's attempts to intercept it. The location we POKE is 14312, so use:

```
POKE 14312,0
```

In fact, we could use POKE to send ALL our graphic codes, but this approach creates timing problems between the printer and computer. Every POKE needs to be followed by a test to see if the printer is ready to accept more codes. Example:

```
10 POKE 14312,0
20 IF PEEK (14312)<>63 THEN 20
```

Line 20 places the program in a delay loop until the printer is ready.

Entering Graphics Mode

0 is also used to enter the graphics mode. For example:

```
10 LPRINT CHR$(27) "K" CHR$(50) CHR$(0);
      ESC      K      NI      N2
```

Since the MX-70 interprets all even numbers as 0, we can change the last part of the above line to

```
      CHR$(2);
```

The POKE alternate is

```
10 LPRINT CHR$(27) "K" CHR$(50); :POKE 14312,0
20 IF PEEK (14312) <>63 THEN 20
```

TRS-80 Model II

The MX-70 works fine with Radio Shack printer cable #26-4401. Be sure the connectors on both ends make good contact. It may be necessary to open the printer case a bit to allow the plug to seat properly.

TRS-80 Model III

No interface kit is necessary for the Model III. Simply plug the Epson cable into the printer port.

The Model III does not share the Model I's problem sending code 0. Nonetheless, another way to send codes to the printer is:

OUT 251,#

where # represents the code we wish to send. If this creates timing problems, we may need to test the status of the printer before sending it each code number. For example:

100 IF PEEK (14312) AND 240 <> 48 THEN 100

places the program in a "holding pattern" until the printer is ready for more. See the computer reference manual if more details are desired.

Using the MX-70 with a Standard Radio Shack Cable

There is a difference between the 'official' Epson printer cable and the ones supplied by Radio Shack. They are wired slightly differently.

The Epson cable allows separation of the CR (carriage return) and the LF (line feed) commands. This in turn allows such things as underlining, overstrikes to slash zeros, sevens, the letter Z, to "black out" material, etc.

If you have no need to separate CR and LF and already have a Radio Shack printer cable, it should work fine. If purchasing a new cable, buy an Epson to keep all the options open.

Appendix E

Use with Atari

Using Atari 800 Computer with Atari 850 Interface

Lines ending with a semicolon cause the Atari to automatically “pad” the rest of the line with spaces until it is 40 characters long, when using LPRINT. Therefore, it is advisable to avoid LPRINT. Fortunately, there is another approach to printing that allows semicolons to be used without problems.

```
10 OPEN #7,8,0, "P"
```

#7 opens tile #7
8 signifies output device
0 is not used
P assigns device as printer

```
20 PRINT #7; "ABC";  
30 PRINT #7; "DEF"  
40 CLOSE #7
```

prints

ABCDEF

on printer.

Port #7 is the printer port usually used by the Atari.

To use the ASCII control codes we often need to add 128. Atari uses codes O-31 for its graphics characters.

If inverse characters are sent as data in the “bit image” mode, the eighth bit will be set, causing the top pin to fire for every character. Highly undesirable!

The Atari 8.50 interface has a 4 second time-out. If it does not receive an acknowledge signal from the printer within 4 seconds of sending it a command, the computer will send a “Device Nak,” Error Code #139 message. The program can catch this error using the BASIC “TRAP” command. Consult your Atari manual for use of TRAP.

Semicolons or commas must be used to separate all literal and variable strings.

The maximum length of a string is 99 characters.

Appendix F

Use with Apple

Apple Integer Basic

The MX-70 is activated by ASCII code numbers from 0-255. The easiest way to send these codes in BASIC is via the CHR\$ function, a feature not supported by Apple Integer Basic.

In theory the codes could all be POKEed from BASIC but it doesn't seem to be a practical option.

Apple Disk Basic

Nearly all programs in this manual are written in Applesoft ROM BASIC. Apple disk users should have no trouble with the programs if the advice in the Apple DOS Manual is heeded. Apple recommends that all DOS commands executed from a BASIC program (ie. with a line number) be preceded by a CTRL-D (CHR\$(4)). For example:

```
10 PRINT CHR$(4) "CATALOG"
```

gives a CATALOG LISTing of all files on the disk.

The DOS commands most frequently needed are PR#1 and PR#0. They can usually be issued from either inside a program (with a line number) or outside the program, at the "command level." In most cases, a simple PR#1 or PR#0 will do fine. In Chapter 3, while we're learning printer control, it is essential they be included INSIDE the program.

There are several ways to issue a CTRL-D from a BASIC program. One is to use CHR\$(4) as in the following example.

```
9 PRINT CHR$(4) "PR#1"
10 PRINT "VIDEO AND PRINTER"
20 PRINT CHR$(4) "PR#0"
30 PRINT "VIDEO ONLY"
```

Many disk users find the CHR\$(4), called "Control D" in lines 9 and 20 unnecessary. They can simply use:

```
9 PR#1
20 PR#0
```

instead. Experiment to see what works best with your system.

Tricky Codes

Most Apple users will encounter difficulty when using codes 9 and 13 in ESCAPE sequences and the graphics mode. To corral these and other rowdy codes, we can POKE the ASCII code numbers directly to the printer driver. Use:

POKE 49296,9

in place of

PRINT CHR\$(9)

Unfortunately, using the POKE statement instead of PRINT creates timing problems between the printer and computer. To avoid such complications, the POKE should always be followed by a test of the printer to see if it is ready for more DATA. On the Apple, this is done with a PEEK statement. Example:

```
10 PR#1
20 POKE 49296,9
30 IF PEEK (49601) <0 GOT0 30
40 PR#0
```

If the value stored at location 49601 (C1C1 Hex) is negative, the printer has not yet picked up the DATA and stored it in its buffer. If location 49601 is positive, the printer is ready to receive more DATA. By using this test we outflank the funny code numbers like 9 and 13.

Screen Width Versus Printer Width

The Apple video screen display is limited to 40 characters per row. Unfortunately, it restricts the printer line width to the same 40 characters during BASIC program listings unless we take evasive action. Fortunately, the solution is simple.

If we send this code sequence down the line:

(Don't do it yet - wait for below)
<CTRL I> 80 N

the printer line width becomes the full 80 columns used by standard printer paper. The "80" can be any number from 40 to 255 as far as the Apple is concerned. The MX-70 might have different ideas.

The prior code sequence also diverts all output to the printer ONLY! The video display is disabled after the CTRL-I # N sequence. The way to bring it back is by:

<CTRL I> I

In the process, the printer width defaults back to 40 columns.

Apple Mush

But how do we send control characters to the printer? The most obvious way is with the CTRL key on the keyboard. Let's try it.

FIRST: activate the printer by typing PR#1 <return>.
SECOND: hold down the CTRL key and press I.
THIRD: type 80N<return>.

Wonderful! Not only does nothing print on the screen, but the printer says ?SYNTAX ERROR.

Not to worry. Try pressing a few keys on the keyboard followed by <return>. The video is disconnected, but the printer works fine. If there is a program in memory, it will LIST fine up to 80 columns wide. To regain control of the video, use the CTRL key again:

<CTRL I>
I <RETURN>

A cleaner way to do the same thing is with the BASIC language CHR\$(9). Try:

PRINT CHR\$(9) "80N"

This kicks us into the printer-only mode without an annoying ?SYNTAX ERROR. To return, use:

PRINT CHR\$(9) "I"

Type carefully as the latter doesn't show up on the screen.

Special Apple Graphics Trick for Long Lines

When printing long lines of graphics, the Apple can be quite finicky. To smooth out the problem, add this line at the beginning of a graphics program, after the PR#1:

9 PRINT CHR\$(9); "255H"

Appendix G

Special Notes on Other Computers/Languages/Interfaces

Interfacing

The MX-70 is designed to interface directly with parallel printer computers like the Apple II and the TRS-80 Models I, II, and III. RS-232, IEEE 448, and other interfaces are not factory supported. There is no provision for installing interface boards inside the case.

Interfacing Cables

The following table identifies known cables which are wired correctly for use with the MX-70:

Computer	Parallel Interface/Cable
Apple II	EPSON I/F P/N 8131 EPSON Cable P/N 8230 'Apple Centronics Parallel I/F
Apple III	Same as above Run in Apple II emulation
Atari 400	2Macrotronics P/N A4P-3
Atari 800	² Macrotronics P/N A8P-3
Atari 400/800 (with 850 I/F)	Macrotronics P/N A85OE
TRS-80 Model I (without expansion I/F)	Radio Shack P/N 26-1411
TRS-80 Model I (with expansion I/F)	Epson Cable P/N 8220 3Radio Shack P/N 26-1401
TRS-80 Model II	Radio Shack P/N 26-4401
TRS-80 Model III	Epson Cable P/N 8220 3Radio Shack P/N 26-1401

'Requires Modification to ground the Most Significant Bit
Kontact Macrotronics for additional information
3Cut Pin #35 at the printer end of the cable

Appendix H

Technical Specifications

1.4.1 Printer

- (1) Printing Method Impact dot matrix
- (2) Character set ASCII 96 (5X 7 dot matrix)
- (3) Character formation,
 - (a) Text mode 5x 7 dot matrix
 - (b) Bit image mode 480 dots/8 inches (horizontal)
- (4) Character size 2.1 mm (W) x 2.8 mm (H) (0.083" x 0.1 1")
- (5) Paper feed Adjustable sprocket feed
- (6) Paper
 - (a) Kind Fanfold paper
 - (b) Width 101.6 mm to 254 mm (4" to 10")
 - (c) Thickness 0.3 mm (0.012") max.
 - (d) Number of copies One original plus one carbon copy
- (7) Line spacing
 - (a) Text mode 1/6 inch
 - (b) Bit image mode 1/9 inch

NOTE: In both modes, line spacing is programmable in increments of 1/72 inch.
- (8) Columns 80 columns (normal size)
40 columns (enlarged size)
- (9) Printing Speed.
 - (a) Text mode 80 characters per second
 - (b) Bit image mode Not defined
- (10) Printing Direction Unidirectional (from left to right). May be returned to home position from any column position.
- (11) Line feed time, Approx. 200 msec (at 116 inch/line)
- (12) Ribbon Cartridge ribbon (exclusive use), Black
- (13) MCBF 5 x 106 lines (excluding print head life)
- (14) Dimensions 374 mm (W) x 305 mm (D) X 107 mm (L)
- (15) Weight Approx. 5 kg

1.4.2 Parallel interface (standard)

- (1) Date transfer rate 1,000 CPS (max),
- (2) Synchronization By externally supplied STROBE pulses.
- (3) Handshaking By ACKNLG or BUSY signals
- (4) Logic level Input data and all interface control signals are compatible with the TTL level.

Appendix I

Control Circuit Board and Initialization

Control circuit board

With the MX-70, such functions as head carriage control, paper feed control, etc., are performed by the one-chip microprocessor 8049 located on the control circuit board. The microprocessor has a 2K-byte ROM and a 128-byte RAM. The print head and two stepper motors are driven by the 16 driver transistors. fig. 1 shows the control circuit diagram

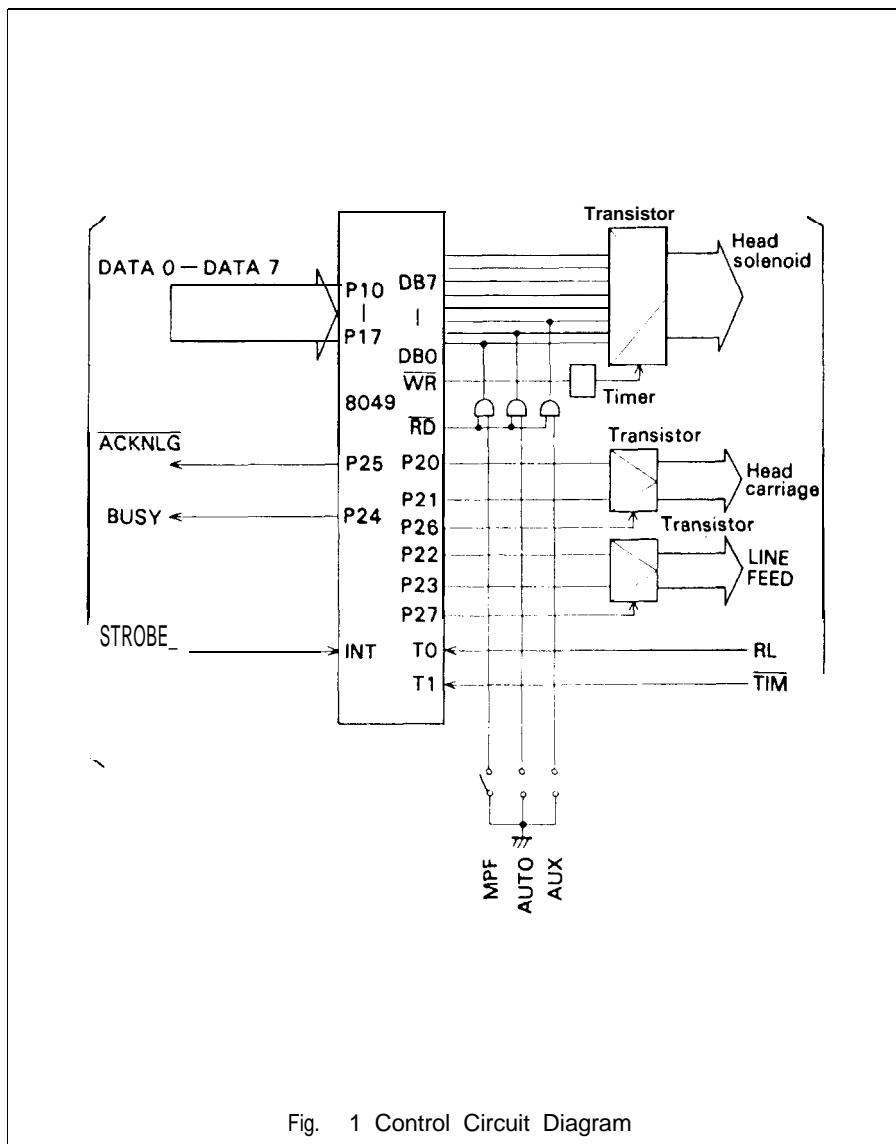


Fig. 1 Control Circuit Diagram

Power circuit

The power circuit generates 5V DC for the logic circuit, and 24V DC to energize the solenoids of the print head and two stepper motors.

Printer initialization

Printer initialization is accomplished in either of the two ways described below.

- (1) Initialization takes place automatically each time the primary AC power source is interrupted and reapplied (i.e., by turning the Power Switch off and on).
- (2) Initialization may be initiated remotely by activating the INIT signal to the parallel interface connector. This line should be driven by a TTL driver or its equivalent.
The minimum INIT pulse width is 50 μ sec at the receiving terminal.
- (3) Upon application of the initialization signal, the following sequence of events takes place in the Printer.
 - (a) The print head returns to its home position.
 - (b) The print buffer is cleared.
 - (c) The line spacing is set at 1/6 inch.
 - (d) The form length per page is set at 11 inches (for the 115V and 240V versions) or 12 inches (for the 220V version).
 - (e) The character print-width logic is reset to the normal state.
 - (f) The operation mode is set in the Text mode.

Appendix J

Pinout Chart

The MX-70 includes a parallel interface as the standard equipment, and this paragraph describes the parallel interface.

(1) Specifications

- (a) Data transfer rate: 1000 CPS (max.)
- (b) Synchronization: By externally supplied STROBE pulses.
- (c) Handshaking: By ACKNLG or BUSY signals.
- (d) Logic level: Input data and all interface control signals are compatible with the TTL level.

(2) Connector

Plug: 57-30360 (AMPHENOL)

It is recommended that interface cables be kept as short as possible

(3) Connector pin assignment and descriptions of signals

Connector pin assignment and descriptions of respective interface signals are provided in Table 7.

Table 7 Connector Pin Assignment and Descriptions of Interface Signals

Signal Pin No.	Return Pin No.	Signal	Direction	Description
1	19	STORAGE	In	<u>STROBE</u> pulse to read data In Pulse width must be more than 0.5 μ s at receiving terminal The signal level is normally "HIGH". read-In of data is performed at the transit to "LOW" level of thus signal
2	20	DATA 1	In	These signals represent information of the 1st to 8th bits of parallel data respectively Each signal is at "HIGH" level when data is logical "1" and "LOW" when logical "0"
3	21	DATA 2	In	The logical state of each data must be held as is until the <u>BUSY</u> signal comes "LOW" level again.
4	22	DATA 3	In	
5	23	DATA 4	In	
6	24	DATA 5	In	
7	25	DATA 6	In	
8	26	DATA 7	In	
9	27	DATA 8	In	
10	28	<u>ACKNLG</u>	out	Approx 5 μ s pulse "LOW" Indicates that data has been received and that the printer is ready to accept other data A "HIGH" signal indicates that the printer cannot receive data The signal becomes "HIGH" in the following cases: 1 During data entry. 2 During printing operation. 3 During paper feed 4 During printer error status

Table 7 (Cont'd)

Signal Pin No.	Return Pin No.	Signal	Direction	Description
12	30	PE	OUT	Always GND Level.
13	-	SLCT	OUT	Pulled up to +5V via 3.3kΩ resistor.
14	-	<u>AUTO</u> FEED X	In	With this signal being at "LOW" level, the paper is automatically fed one line after printing. (The signal level can be fixed to "LOW" with AUTO provided on the control circuit board.)
15	-	NC		Not used.
16	-	OV		Logic GND level.
17	-	CHASSIS	GND	Printer chassis GND. In the printer, the chassis GND and the logic GND are isolated from each other.
18	-	+5V	-	+5 V
19 to 30	-	GND	-	TWISTED-PAIR RETURN signal GND level.
31	-	<u>INIT</u>	In	When the level of this signal becomes "LOW", the printer controller is reset to its initial state and the print buffer is cleared. This signal is normally at "HIGH" level. and its pulse width must be more than 50μs at the receiving terminal.
32		ERROR	Out	Always "HIGH" level Pulled up to +5V via 3.3kΩ resistor.
33	-	GND	-	Same as with Pin Nos. 19 to 30.
34	-	RESET	Out	Reset signal generated when the Printer is powered, or when the INIT signal is applied to the Printer.
35				Pulled up to +5V via 3.3kΩ resistor.
36	-	NC	-	Not used.

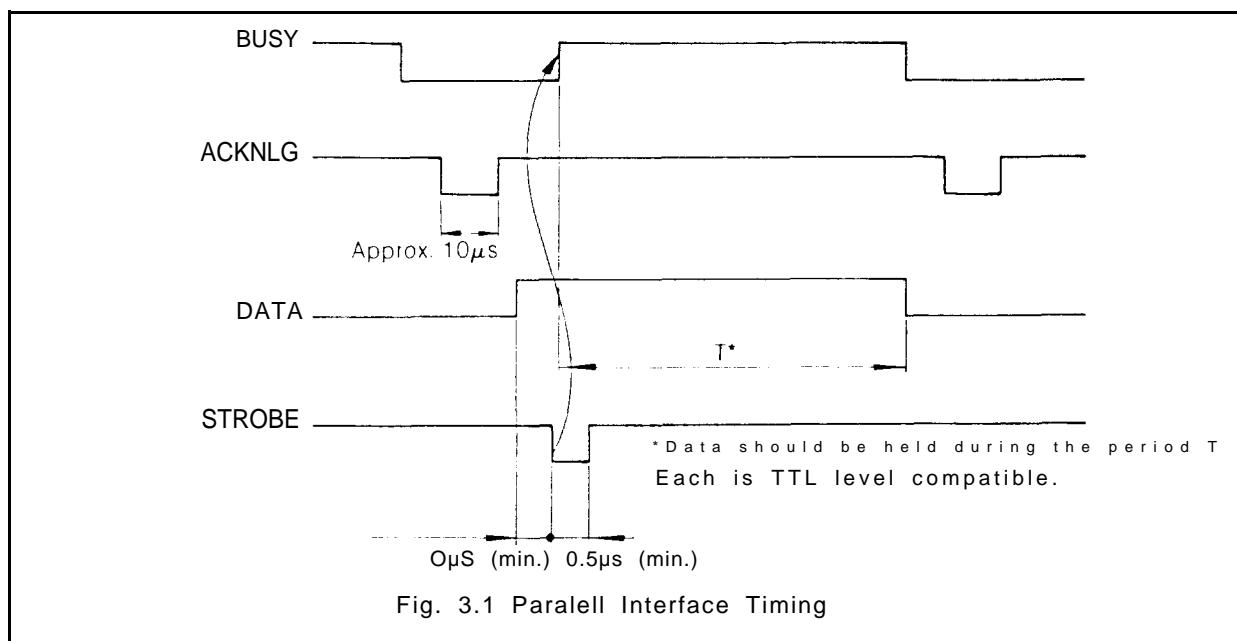
NOTES:

1. "Direction" refers to the direction of signal flow as viewed from the printer
2. "Return" denotes "TWISTED PAIR RETURN" and is to be connected at signal ground level
As to the wiring for the Interface, be sure to use a twisted-pair cable for each signal and never fail to complete connection on the Return side. To prevent noise effectively, these cables should be shielded and connected to the chassis of the host computer and the printer, respectively.
- 3 All interface conditions are based on TTL level. Both the rise and fall times of each signal must be less than 0.2μs.
4. Data transfer must not be carried out by ignoring the ACKNLG or BUSY signal. (Data transfer to this printer can be carried out only after confirming the ACKNLG signal or when the level of the BUSY signal is "LOW".)

Appendix K

Parallel Interface Timing

The MX-70 includes a parallel interface as standard equipment, which is compatible to the Centronics parallel interface. The Printer can be connected easily to any computer with this interface structure. Fig. 3.1 shows the timing chart when signals are to be input from an external device to the MX-70. As is apparent from the timing chart, when data are transmitted to data lines DATA 1 through DATA 8 and the STROBE signal is generated by an external device, the MX-70 sets the BUSY signal to "HIGH" level to inhibit the next data transfer from the external device. The logical state of each data must be held as is until the BUSY signal becomes "LOW" level again, after the transmission of the STROBE signal from the external device, or Incorrect data may be written in the print buffer of the MX-70.



This timing chart also reveals that the ACKNLG signal from the MX-70 can be used in place of the STROBE signal. This allows the Printer functions to be checked without use of a special Instrument, in such a manner as described below.

STEP 1. Connect pin No. 1 (STROBE) to Pin No. 10 (ACKNLG) with a lead.

STEP 2. Connect pin Nos. 2 through 9 (DATA 1 through 8) of the interface connector as required according to the character to be printed. For example, if character "A" (represented by <41>H in ASCII code) is to be printed, connect pin Nos. 3, 4, 5, 6, 7 and 9, respectively to the ground (e.g., pin No. 19). Note that when pin Nos. 2 through 9 are left open without connecting them to the ground, data signals become "HIGH" level and when they are connected to the ground, data signals become "LOW" level.

STEP 3. Connect the jumper No. 2 (AUTO) on the control circuit board to enable the AUTO FEED function.

By the above operating steps, the MX-70 operations including the interface can be checked. Unlike the self-test (self-diagnostic) function by the built-in program, this method permits the Printer operations to be checked along with the parallel interface operation.

Appendix L

Printer Maintenance

A clean and comfortable environment will insure the best possible service from your printer (and its operator).

Find it a Nice Home

Your Epson printer is designed to take a lot of punishment, but there is no reason to become careless. In fact, it will last a lot longer if you pamper it. Treat it at least as good as your pet rock.

- Avoid operation in direct sunlight.
- Avoid use in areas where there is a high concentration of dust or grease.
- Avoid humid locations and heat sources such as a furnace.
- Never install it near large electrical motors.
- Your printer (like your computer) likes a nice comfortable range of temperatures. The safe operating range is 5°C (40°F) to 35°C (95°F).

Periodic Maintenance

Clean it periodically with a soft brush to remove paper and dust particles. The exterior can be cleaned with a mild detergent and water or denatured alcohol. The interior can be cleaned with denatured alcohol.

The ribbon cartridge has an expected life of 3 million characters. Replacements are readily available through your dealer.

Epson provides a technical manual to assist do-it-yourselfers in really gumming up the works. Contact your dealer for price and availability.

Changing the Print Head

The print heads expected life is 100,000,000 characters. Readers who are easily diverted by trivia can estimate the average number of characters printed per line and the average number of lines printed per day. Multiply them together

and divide **into** 100,000,000. The result is the estimated print head life in days. (Then count the legs and divide by four . . .)

If and when the print head needs replacement, make sure it is cool to the touch. It may operate very **hot** under heavy use.

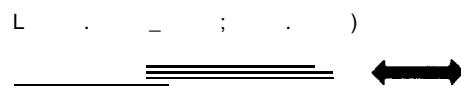
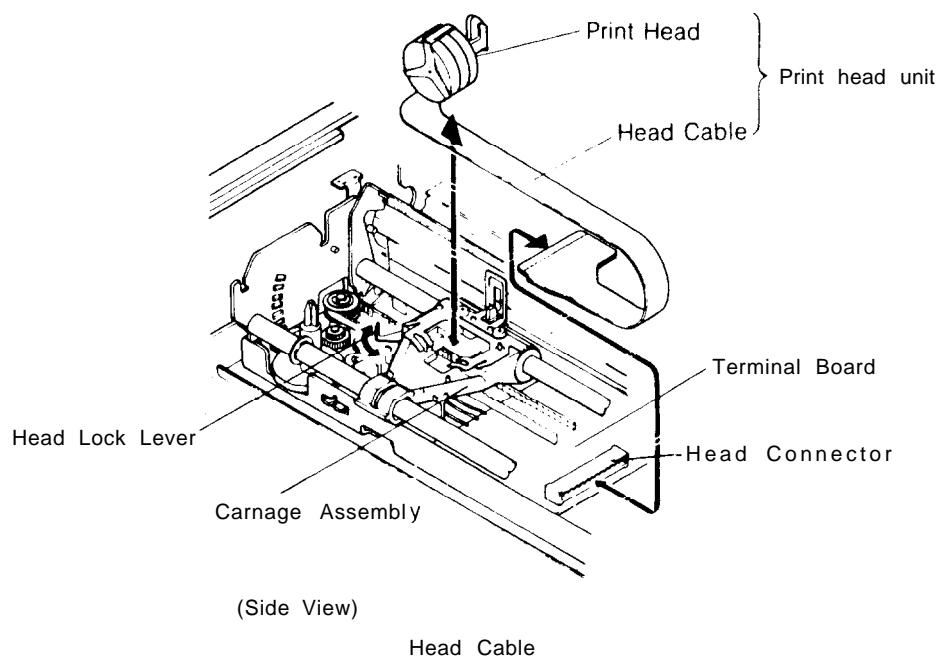
Remove the printer lid and ribbon cartridge. Turn the head lock lever clockwise and pull the print head straight up and off.

Don't pull too far! It's still connected to the base of the printer by a flat ribbon cable. Grasp the ribbon cable (not the head connector block!) and pull gently but firmly to your right.

You have just removed your first Epson print head. (Shades of the French Revolution.)

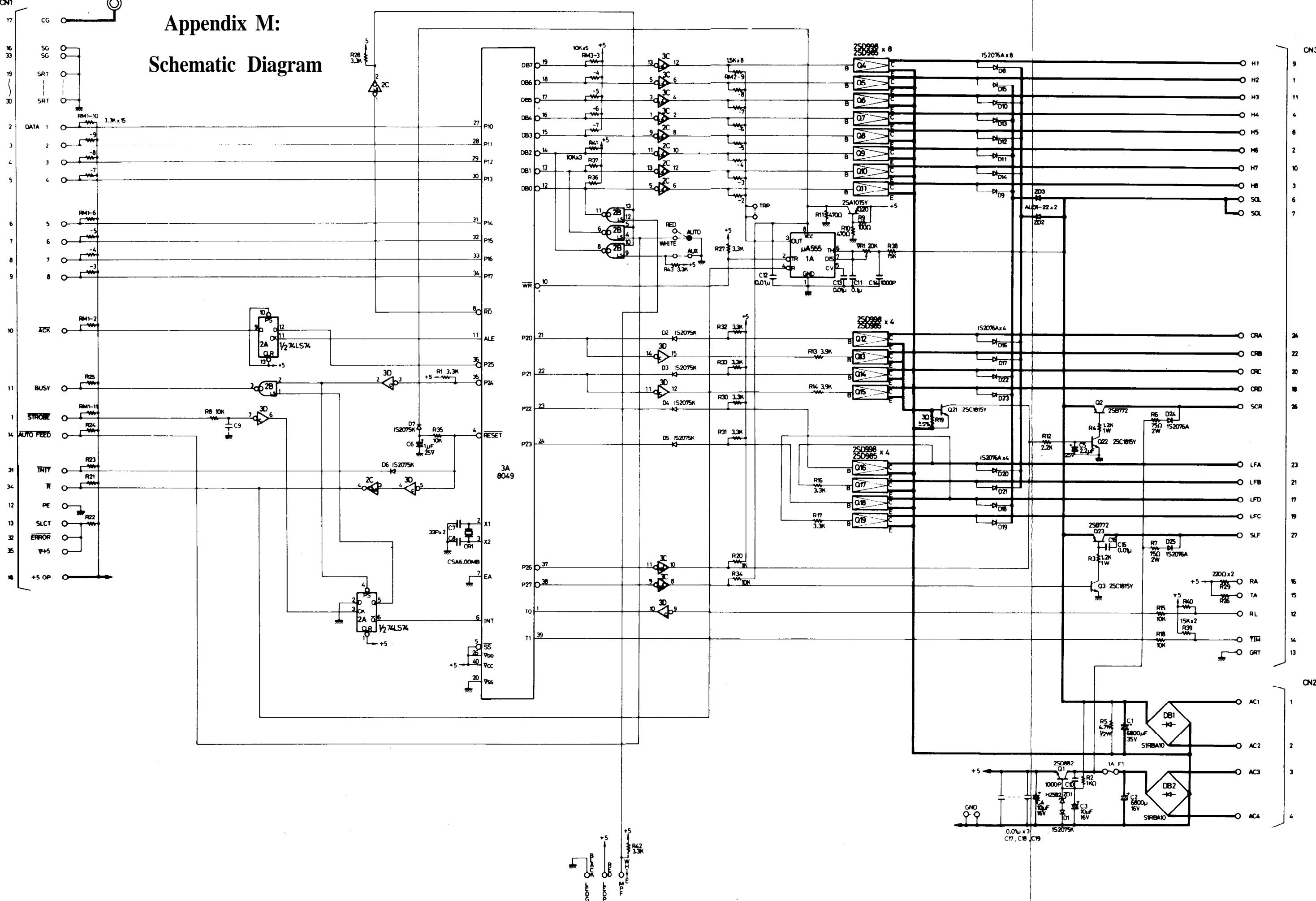
To install a new print head assembly, push the new ribbon cable into the head connector, lay the print head on its mount and turn the head lock lever counterclockwise.

Voila! Back in business. And it took less time than solving the Rubik cube.



*Take hold of the cable at the point indicated by arrows
8 and apply force in either of the directions indicated
by arrow c) to push in or pull out the head cable.

Appendix M: Schematic Diagram



INDEX

	Page
Automatic line feed, proper setting	4, 6, 16
Cable, printer	1, 7, 87
Chararterset	83, 84
CHR\$	13, 18, 23, 31
Codes	
ASCII	13, 21, 79, 80
Control	13, 81, 91, 93
Escape	19
Problem	20, 37, 42, 45, 55, 85, 92
Dot matrix	31
Double width printing	19, 23
EXCHANGE TIMES	4
Feed button	11
Form Feed command	26, 27
Form length - see Page length	
Graphics	
High resolution	14, 67
Mode	20, 41, 62, 86
Pin labels	42, 43
Programexamples	46, 64
Horizontaltabs	28
Line Feedcommand	37, 87
Linespacing	34, 36, 49
LIST	18
LLIST	18, 85
LPRINT	15, 16, 85, 89
Maintenance	105
Pagelength	24, 25, 28
Paperfeeding	8, 9, 10
Paper standards	24
Paper thickness adjustment	10
PEEK	52, 53, 86, 87, 92
Pinoutsignals	101
POKE	14, 37, 52, 53, 55, 86, 92
Power	
Light	10, 11
Requirements	10
Switch..	10
PR#(Apple)	14, 23, 91
Print head..	105
Printer	
Initialization	15
Interfacing	1, 4, 95
Rack,paper	8
Ribbon installation	3
Screendump(Apple)	14, 63, 67
Selftest	11
Seven bit limitation	31, 45, 49, 61, 63
Shippingscrews	2
Tractorfeed..	8
Topofform9, 25, 3
Underlining	37
USR	74
Welcome program	17, 46

NOTICE

**Please send your comments, suggestions, and any errors
you might find to:**

**THE EDITOR
MX-70 Manual
60x 19669
San Diego
California 92119
U.S.A.**

Thank You.

LIMITED WARRANTY

Epson America, Inc., "Epson," warrants this product against defects in material and workmanship for a period of ninety (90) days from the original date of purchase by only the original consumer purchaser. This Warranty applies only should the product fail to function properly under normal use.

During the warranty period, Epson will repair (or, at its option, replace), at no charge, components that prove to be defective, provided the product is returned (shipping prepaid and properly packed) to the nearest authorized Epson Service Center or to the Dealer from whom the product was purchased.

You may request information on how to obtain service by contacting the Epson dealer from whom the product was purchased, or by contacting Epson directly at the address printed below.

Your purchase receipt or other proof of date of original consumer purchase and the serial number of the product must be provided when requesting warranty service to be performed. This warranty does not apply if the Epson label or rating label or the serial number has been removed, defaced, or altered, or if, in the opinion of Epson, the product has been damaged by accident, misuse, neglect, improper packing or improper modifications or service by other than an authorized Epson Service Center.

This warranty is in lieu of all other express warranties, statements or representations, and unless stated herein, all such warranties, statements or representations made by any other person or firm are void. All implied warranties in connection with this sale, including the warranty of merchantability, shall be of the same duration as the warranty period stated above.

In the event that this product shall prove defective in workmanship or materials, your sole remedy shall be the repair or remedy as stated in this warranty, and under no circumstances shall Epson be liable for any loss or damage, direct, incidental or consequential, arising out of the use of, or inability to use, this product caused by any defect in the product. Some states do not allow the exclusion or limitation of incidental or consequential damages, nor permit limitations of the duration of any implied warranty, so the above limitations or exclusions may not apply to you. **THIS WARRANTY APPLIES ONLY WHEN THIS PRODUCT IS PURCHASED AND USED SOLELY IN THE UNITED STATES OF AMERICA.**

Epson America, Inc.
Service Center
3415 Kashiwa Street
Torrance, CA 90505

(213) 539-9140